T.C.
İSTANBUL UNIVERSITY
INSTITUTE OF GRADUATE STUDIES IN
SCIENCE AND ENGINEERING

M.Sc. THESIS

ON DESIGNS DERIVED FROM CODES

Kübra DURAN

Department of Mathematics

Mathematics Programme

SUPERVISOR
Assoc. Prof. Dr. Şükrü YALÇINKAYA

CO-SUPERVISOR
Assoc. Prof. Dr. Fatih DEMİRKALE

October, 2017

İSTANBUL

This study was accepted on 24/11/2017 as a M. Sc. thesis in Department of Mathematics, Mathematics Programme by the following Committee.

Examining Committee Members

Assoc. Prof. Dr. Şükrü YALÇINKAYA(Supervisor)
İstanbul University
Faculty of Science

Prof. Dr. Kamuran SAYGILI
Istanbul University
Faculty of Science

Assoc. Prof. Dr. Kadri Ulaş AKAY
Istanbul University
Faculty of Science

Assoc. Prof. Dr. Emre KOLOTOĞLU
Yıldız Technical University
Faculty of Science

Assist. Prof. Dr. Cemal ÇİÇEK
Istanbul University
Faculty of Science

As required by the 9/2 and 22/2 articles of the Graduate Education Regulation which was published in the Official Gazette on 20.04.2016, this graduate thesis is reported as in accordance with criteria determined by the Institute of Graduate Sttudies in Science and Engineering by using the plagiarism software to which İstanbul University is a subscriber.

FOREWORD

I would like to begin with to indicate my extensive gratitude to my thesis supervisor Assoc. Prof. Dr. Şükrü YALÇINKAYA for his attentive working on my thesis, for his enlightening comments and guidance, and for his patience throughout this process. Also, I owe my thanks to my thesis co-supervisor Assoc. Prof. Dr. Fatih DEMİRKALE for his refinement and temperate approaches during our studies.

I would like to thank my thesis committee.

I thank TÜBİTAK for the financial support.

I owe my thanks to my father Sinan YÜCEL, my mother Hatice YÜCEL, my sister Esra and my brother Abdurrahim for their unending love, patience and supports during my education and my life.

I would like to thank my father in law Assoc. Prof. Dr. Ahmet DURAN and my mother in law Şerife DURAN for their love and powerful support. I also thank brothers in law Mehmet, Cemil and Mümin for their supports and motivation.

Finally, I would like to express my deepest gratitudes to my husband Burhan for his love and endless supports. During this work, he has motivated and encouraged me to do my best.

November 2017 Kübra DURAN

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Explanation
$\boldsymbol{B_e(\bar{x})}$	: The ball of radius e around a word $\bar{x}$
$\boldsymbol{\lvert C \rvert}$	: The size of a code C
$\boldsymbol{\bar{C}}$	: The extended code of a code C
$\boldsymbol{C^{\perp}}$	: The dual of a code C
$\boldsymbol{d(C)}$	: The minimum distance of a code C
$\boldsymbol{d_H(\bar{x},\bar{y})}$	: The Hamming distance between $\bar{x}$ and $\bar{y}$.
$\boldsymbol{F_q^n}$	: The set of all n-tuples from q-symbol alphabet.
$\boldsymbol{(n,M,d)}$	: A code of length n, size M and minimum distance d
$\boldsymbol{[n,M,d]}$	: A linear code of length n, size M and minimum distance d
$\boldsymbol{S_e(\bar{x})}$	: The sphere of radius e around $\bar{x}$
$\boldsymbol{S_\lambda(t,k,v)}$	: A Steiner system
$\boldsymbol{S(t,k,v)}$	: A Steiner system with $\lambda = 1$
supp($\bar{x}$)	: The support of $\bar{x}$
$\boldsymbol{t\text{-}(v,k,\lambda)}$	: A t-design
$\boldsymbol{w_H(\bar{x})}$	: The Hamming weight of $\bar{x}$
$\boldsymbol{w(C)}$	: The minimum weight of a code C
$\boldsymbol{W_C(z)}$	: The weight enumarator of a code C

ÖZET

YÜKSEK LİSANS TEZİ

KODLARDAN ÜRETİLEN TASARIMLAR ÜZERİNE

Kübra DURAN

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Matematik Anabilim Dalı

Danışman : Doç. Dr. Şükrü YALÇINKAYA
II. Danışman : Doç. Dr. Fatih DEMİRKALE

Bu tezde, kodlar, kombinatoryal tasarımlar ve kodlar ile kombinatoryal tasarımlar arasındaki ilişkiler çalışılmaktadır. Mathematica'da boyutu ve uzunluğu, sırayla, 14 ve 15'e kadar olan bütün lineer kodları listeleyen ve bu kodlar içinde 1-tasarım oluşturan kodsözcüklerinin ağırlıklarını bulan bir program hazırlanmıştır. Bu programdan alınan veriler kullanılarak, 1-tasarımlarla ilgili bazı genel sonuçlar ve ispatları verilmektedir.

Bu tezin ilk bölümünde Kodlama Teorisi ve Kombinatoryal Tasarım Teorisi ile ilgili temel tanımlar ve bu teorilerin tarihsel gelişimleri verilmektedir. İkinci bölümün ilk kısmında kodlama teorisindeki temel terminolojiler tanıtılıp kodlarla ilgili örnekler ve temel teoremler de verilmektedir. İkinci kısımda, kombinatoryal tasarımlar tanıtılıp örnekler verilmektedir ve bir kombinatoryal tasarımın cebirsel gösterimi verilmektedir. Üçüncü kısımda kodlar ve kombinatoryal tasarımlar arasındaki ilişkilerden bahsedilmektedir, sonra iyi bilinen Fano düzlemi örnek olarak verilmektedir. Yine bu kısımda Assmus-Mattson teoremleri tanıtılmaktadır. Üçüncü bölümün ilk kısmında, boyutu ve uzunluğu sırasıyla 14 ve 15'e kadar olan değerler için programımızdan elde edilen sayısal sonuçlar tablolar halinde verilmektedir. İkinci kısımda 'tasarım gömülü kodlar' tanıtılmaktadırr. Bu kısımda küçük uzunluk ve boyut için linear tasarım gömülü kodları ve 1-tasarım oluşturan kodsözcüklerin ağırlıklarını listeleyen programımızdan alınan verilere göre belirlenen genel sonuçlar ve bunların ispatları verilmektedir.

Kasım 2017, 70 sayfa.

Anahtar kelimeler: Kodlar, Kombinatoryal Tasarımlar, Tasarım Gömülü Kodlar

SUMMARY

M.Sc. THESIS

ON DESIGNS DERIVED FROM CODES

Kübra DURAN

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Mathematics

Supervisor : Assoc. Prof. Dr. Şükrü YALÇINKAYA
Co-Supervisor : Assoc. Prof. Dr. Fatih DEMİRKALE

In this thesis, we study codes, combinatorial designs and the relations between codes and combinatorial designs. On Mathematica, we write a program which lists all linear codes of dimension and length up to 14 and 15, respectively, and finds the weigths of all codewords forming 1-designs in these codes. By using the data obtained from the program, we present some general results and their proofs about 1-designs.

In the first chapter of this thesis, we present the fundamental definitions in Coding Theory and Combinatorial Design Theory and also their historical backgrounds. In the first section of the second chapter, we introduce basic terminologies in coding theory, and present examples and also fundamental facts on codes. In the second section, we introduce combinatorial designs, present examples and the algebraic representation of a combinatorial design. In the third section, we establish the connections between codes and combinatorial designs, then we give the well-known Fano Plane as an example. Moreover, in this section, we introduce Assmus-Mattson Theorems. In the first section of the third chapter we give the numerical results obtained from our program in tables for the values of the dimension and the length up to 14 and 15, respectively. In the second section, we introduce 'design embedded codes'. In this section, we present general results that we infer from our program, which lists all linear 1-design embedded codes and the weights of all codewords forming 1-designs in these codes, for small values of length and dimension and the proofs of these results.

November 2017, 70 pages.

Keywords: Codes, Combinatorial Designs, Design Embedded Codes

1. INTRODUCTION

Coding theory has been an active research area in mathematics since Shannon's paper [1]. The main problem of coding theory is to detect or correct errors in transmitted messages. Here, transmission means sending messages from one place to another, i.e. communication, or saving messages from the present time to future.

In [1], Shannon expressed and proved that a reliable communication in the desired quantity can be occured if appropriate techniques of encoding and decoding are used over a channel capacity. However Shannon's proof is not constructive, that is, it doesn't include any encoding and decoding method which can be used. Later, Richard W. Hamming [2] published a paper entitled, *'Error Detecting and Correcting Codes'*. After this work, developments in coding theory build rapidly.

History of combinatorial design theory dates back to a long time in mathematics. It concerns about arranging elements of a finite sets into subsets according to specified rules. It can be said that the history of combinatorial design theory goes back to 1782, Euler's 36 officers problem. Küçükçiftçi and Özkan [3] have worked on the question about 2-designs asked by Wesley Woolhouse in 1844. And also in 1850, Kirkman [4] gave the Kirkman's schoolgirl problem which is one of the earliest examples of combinatorial designs. More information about the history of combinatorial designs is, for example, presented in [5]. As in the above examples of design theory, combinatorial designs were a consequence of mathematical curiosity in earlier. Nowadays combinatorial designs are used in most areas such as computer science [6], cryptography, communications, networking [7], food research [8], DNA sequencing [9], more information can be found in [10].

In 1966, Alltop[11] presented a method for obtaining t-designs, mostly $t \geq 2$. We will introduce t-designs in Section 2.2.

We organize this thesis in five parts; Introduction, Materials and Methods, Results, Discussion, Conclusion and Recommendations.

Chapter 2, Materials and Methods, has three sections and we give the main subjects of the thesis in this chapter. In the first section, we give a short overview of some of the basic notations and concepts of algebraic coding theory. We begin with introducing a code and its structures, code alphabet and code symbol. Then we continue with the examples on codes. We introduce binary codes, ternary codes and give related examples. After that, we introduce linear codes which is a very important subject for coding theory. We give definitions, properties and examples and then give some relations between parameters of a code. We also present error detecting and correcting codes. And then we give the definition of Hamming codes. Then, in the second section, we present a brief overview of some of the basic notations and concepts of the combinatorial design theory. We begin with definition of a design, its notations. We give some examples and theorems about designs. In this section we also give incidence matrix of a design. In the third section, we establish the connections between codes and designs, and the support of a code. We give a famous example, Fano plane, to explain the relation between codes and designs. We give Assmus-Mattson Theorems, as these theorems are important in finding t-designs from codes.

We prepared a computer program by using Mathematica which lists all the generating matrices for linear codes and finds the weigths of all codewords forming 1-designs in these codes. We present the program in Appendix. In Chapter 3, we present the numerical results of this program in tables for length and dimension less than 15 and 14, respectively. In Section 3.2 we present 'design embedded codes'. To obtain designs from codes is rather difficult than to obtain codes from designs since designs require more restrictive conditions. In this section, according to the results obtained from the program for small values of length and dimension, we put necessary conditions on the general values of n and k. Then, we present some results and their proofs about 1-design embedded codes.

In Chapter 4, we give the concluding remarks.

In Chapter 5, we review the study. We examine all chapters, their contents and discuss what can be done for a future study.

2. MATERIALS AND METHODS

2.1. BACKGROUNDS ON CODES

We give a short overview of some of the basic notations and concepts of algebraic coding theory. The proofs of the results which are presented in this chapter and more details about algebraic coding theory can be found in [12] and [13].

Let $\mathbb{F}_q$ be a finite field with q elements. In this thesis, we use the notation $\mathbb{F}_q^n$ where $\mathbb{F}_q$ represents a q-symbol alphabet and the elements of $\mathbb{F}_q^n$ are n-tuples over $\mathbb{F}_q$. Note that, $\mathbb{F}_q^n$ is an n-dimensional vector space over $\mathbb{F}_q$.

A code of length n (or a q-*ary code*) over $\mathbb{F}_q$ is defined to be a nonempty subset $C \subseteq \mathbb{F}_q^n$.

Let we have a set with q elements such $A = \{a_1, a_2, \dots, a_q\}$. We call the set A as our *code alphabet*, each of a_i as *code symbol*. We take the field $\mathbb{F}_q$ as code alphabet and the elements of the field $\mathbb{F}_q$ as code symbols. *A word of length* n is just a juxtaposition of n code symbols. *Codeword is* said for the elements of a code C and they are symbolized by $\bar{c} \in C$. The codewords can be thought as vectors in a vector space $\mathbb{F}_q^n$.

Example 2.1.1. Let $q = 3$. Then, 012010 is a word of length 6.

Example 2.1.2. Let C_1 be a code over $\mathbb{F}_2$ of length 3 such that $C_1 = \{000, 101, 111\}$. Then $000, 101$ and 111 are codewords of C_1.

The cardinality of a code C is called *the size of the code*, so it is denoted by $|C|$.

Example 2.1.3. Let $C_2 = \{10000, 20101, 11122, 22222\}$ be a code over $\mathbb{F}_3$ of length 5.Then, the size of the code is $|C_2| = 4$.

Assume C be a code over $\mathbb{F}_q$. If $|C| = M$ and C is of length n, C is called an (n, M)-*code*.

Example 2.1.4. Assume that C_1 and C_2 are two codes from Example 2.1.2 and Example 2.1.3, respectively. Then, C_1 is $(3,3)$-code and C_2 is $(5,4)$-code.

Definition 2.1.5. When $q = 2$, we call the codes over $\mathbb{F}_2$ as *binary*, which are the most common type that appears in applications. When $q = 3$, we call the codes over $\mathbb{F}_3$ as *ternary*. We call the codes as *quaternary* when $q = 4$.

There are some important advantages of using binary codes in applications. Firstly, the set $\{0,1\}$ together with ordinary addition and multiplication is the smallest finite field and to correct the error it is enough to locate the error. We introduce error detection and error correction in the Definition 2.1.35 and Definition 2.1.37, respectively.

Now, we shall define linear codes which are a significant class of codes.

Definition 2.1.6. A *linear code* which is of length n over the field $\mathbb{F}_q$ is defined to be a vector subspace of $\mathbb{F}_q^n$.

A linear code of length n over $\mathbb{F}_q$ is a k-dimensional vector space of $\mathbb{F}_q^n$. Then, it is indicated by $[n,k]$-*code*. Therefore, *a basis of a linear code* is described as a basis of the corresponding vector space. Observe that, any $[n,k]$-code has a basis which consists of k codewords. Thus, an $[n,k]$-code has size q^k. If $B = \{\bar{c_1}, \bar{c_2}, \dots, \bar{c_k}\}$ is a basis of a code C, then

$$C = \left\{\alpha_1\bar{c_1} + \alpha_2\bar{c_2} +, \dots, +\alpha_k\bar{c_k} : \alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{F}_q \right\} \tag{2.1}$$

Example 2.1.7. Assume that $B_1 = \{100, 010, 001\}$ is a basis of a linear code S over the field $\mathbb{F}_2$, then $S = \{000, 100, 010, 001, 110, 101, 011, 111\}$.

Now, we present the definitions of the generating matrix and parity check matrix which are the familiar representations of a linear code.

Definition 2.1.8. Assume that C is a linear code. *A generating matrix G* for this code is a matrix whose rows forms a basis for C.

Example 2.1.9. Assume that C is a linear code over $\mathbb{F}_2$ and $G = \begin{bmatrix} 1001 \\ 0101 \end{bmatrix}$ is a generating matrix for this code. Therefore, all codewords of C is the set $\{\alpha_1[1001] + \alpha_2[0101] : \alpha_1, \alpha_2 \in F_2\}$. Observe that, $C = \{0000, 1001, 0101, 1100\}$.

Definition 2.1.10. Consider that C is an $[n, k]$-*code* over $\mathbb{F}_q$. For this code, the parity check matrix is the $(n-k) \times n$ matrix H which satisfies

$$C = \{\bar{x} \in \mathbb{F}_q^n \,|\, H\bar{x}^T = \bar{0}\} \tag{2.2}$$

In general, linear $[n, k]$-codes have many generating matrices. If the first k columns are independent then the generating matrix for a linear code is in standart form, that is $[I_k|A]$, where I_k is the $k \times k$ identity matrix and A is an $k \times (n-k)$ matrix. A linear code is a kernel of some linear transformation since it is a subspace of a vector space. Therefore, since parity check matrices are kernels of some linear transformations, they are not uniquely determined for linear codes. Now, we state the relation between generating matrix and parity check matrix.

Theorem 2.1.11. Assume that $G = [I_k|A]$ is a generating matrix for the $[n, k]$-code C, therefore the corresponding matrix $H = [-A^T|I_{n-k}]$ is a parity check matrix for C, where $A^T$is the transpose of A.

Proof. Observe that $\bar{x} \longmapsto H\bar{x}^T$ is a linear transformation. Now, we show that $HG^T = 0$ and $\dim(\ker(\bar{x} \longmapsto H\bar{x}^T)) = k$, which is equal to the dimension of C. Since A is a $k \times (n-k)$ matrix, we get a $(n-k) \times k$ matrix,

$$HG^T = [-A^T|I_{n-k}][I_k|A]^T = -A^T I_k + I_{n-k}A^T = 0.$$

Thereby, the kernel of this linear transformation contains the code C. Because rank(H)= $n-k$, $\dim(\ker(\bar{x} \longmapsto H\bar{x}^T)) = k$, which is equal to the dimension of C. The result follows. □

Example 2.1.12. Assume that C is a linear code over $\mathbb{F}_2$ with generating matrix $G = \begin{bmatrix}1001\\0101\end{bmatrix}$. Then we find the parity check matrix H for C. Here, we have the matrix $A = \begin{bmatrix}01\\01\end{bmatrix}$ since the generating matrix for C is $G = [I_2|A]$. Also the code is of length 2 and dimension 2. Then $-A^T = \begin{bmatrix}00\\11\end{bmatrix}$. So, $H = \begin{bmatrix}0010\\1101\end{bmatrix}$.

Next, we give an example to find out a code from a parity check matrix.

Example 2.1.13. For a linear code C over $\mathbb{F}_2$, let $H = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. We will find the corresponding binary code C. Then,

$$C = \left\{ (x_1 x_2 x_3 x_4) \in \mathbb{F}_2^4 \mid H \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}.$$

So,

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{implies} \quad x_1 + x_3 + x_4 = 0, \quad x_2 + x_4 = 0, \quad x_3 + x_4 = 0.$$

Then they are found as $x_1 = 0$, $x_2 = x_3 = x_4 = r$, for all $r \in \mathbb{F}_2$. Then, we get the code $C = \{0rrr \mid r \in \mathbb{F}_2\} = \{0000, 0111\}$.

Definition 2.1.14. The dual of C, $C^{\perp}$, is defined by the following set

$$C^{\perp} := \{\bar{x} \in \mathbb{F}_q^n \,|\, \forall \bar{y} \in C, \langle \bar{x}, \bar{y} \rangle = 0\} \tag{2.3}$$

where $\langle\ ,\ \rangle$ is the standart inner product in $\mathbb{F}_q^n$.

Here, codewords can be thought as vectors in a vector space $\mathbb{F}_q^n$.

Theorem 2.1.15. Even if C is not a linear code, the dual of this code, $C^{\perp}$, is always linear.

Proof. For the linearity, we have to show that if $\bar{u}, \bar{v} \in C^{\perp}$ and for all $a, b \in \mathbb{F}_q$, $a\bar{u} + b\bar{v}$ is in $C^{\perp}$. Let $\bar{u}, \bar{v}$ be two codewords in $C^{\perp}$ and $\bar{z}$ be a codeword in C, from the duality we have $\langle \bar{u}, \bar{z} \rangle = 0$ and $\langle \bar{v}, \bar{z} \rangle = 0$ for all $\bar{z} \in C$. Then, by the properties of the inner product, we have $\langle a\bar{u}, \bar{z} \rangle = 0$ and $\langle b\bar{v}, \bar{z} \rangle = 0$ and also $\langle a\bar{u} + b\bar{v}, \bar{z} \rangle = 0$. Hence, $a\bar{u} + b\bar{v} \in C^{\perp}$. □

Theorem 2.1.16. If the code C is an $[n, k]$-linear code over $\mathbb{F}_q$ with generating matrix G and the parity check matrix H, then $C^{\perp}$ is an $[n, n-k]$-linear code with the generating matrix H and the parity check matrix G.

Proof. Let $G = [I_k | A]$ and $H = [-A^T | I_{n-k}]$. Let $\bar{x} \in C$ and $\bar{y} \in C^{\perp}$, then for all $\bar{x}$ and $\bar{y}$, the inner product $\langle \bar{x}, \bar{y} \rangle = 0$ from duality. Since all elements of C is represented by the generating matrix G, we can write $G\bar{y}^T = 0$, for all $\bar{y} \in C^\perp$ instead of the inner product. So it means, $C^\perp = \{\, \bar{y} \in \mathbb{F}_q^n \mid G\bar{y}^T = 0\}$ and therefore, for $C^\perp$, G be a parity check matrix. Since $C = \{\bar{x} \in \mathbb{F}_q^n | H\bar{x}^T = 0\}$, every row of H is in the dual $C^\perp$. Now, to say that H is a generating matrix for $C^\perp$, so we need to show that $dim(C^\perp) = rank(H)$. Here, $rank(H) = n - k$ since there are $n - k$ linearly independent column vectors of H, we have $dim(C^\perp) = n - rank(G) = n - k$. □

Definition 2.1.17. Consider a code C be linear and $C^\perp$ is a dual code of this code. If the code and its dual equal to each other, $C = C^\perp$, then this code is called *self-dual*. C is called *self-orthogonal* if $C \subseteq C^\perp$.

Example 2.1.18. Let $S = \{0101, 1010\}$ be a code over $\mathbb{F}_2$ and let $\bar{v} = (v_1 v_2 v_3 v_4) \in S^\perp$. Then,

$$\langle \bar{v}, (0,1,0,1)\rangle = 0 \Rightarrow v_2 + v_4 = 0,$$

$$\langle \bar{v}, (1,0,1,0)\rangle = 0 \Rightarrow v_1 + v_3 = 0.$$

Thereby, we get $v_2 = v_4$ and $v_1 = v_3$. Here, v_1 and v_2 can be either 0 or 1. Then, we get the dual code as $S^\perp = \{0000, 0101, 1010, 1111\}$. So, S is a self-orthogonal code. Note that S is not a linear code and here we see that $S^\perp$ is linear, see Theorem 2.1.15.

Example 2.1.19. Consider C be an [8,4]-linear code with the following generating matrix $G = \begin{bmatrix} 1&1&0&1&0&0&0&1 \\ 1&0&1&0&1&0&0&1 \\ 0&1&1&0&0&1&0&1 \\ 1&1&1&0&0&0&1&0 \end{bmatrix}$. Let us check if C is a self-orthogonal code. Let $\bar{r_i}$ denote the rows of G. Then, $\langle \bar{r_i}, \bar{r_i}\rangle = 4$, $\langle \bar{r_1}, \bar{r_2}\rangle = 2$, $\langle \bar{r_1}, \bar{r_3}\rangle = 2$, $\langle \bar{r_1}, \bar{r_4}\rangle = 2$, $\langle \bar{r_2}, \bar{r_3}\rangle = 2$, $\langle \bar{r_2}, \bar{r_4}\rangle = 2$, $\langle \bar{r_3}, \bar{r_4}\rangle = 2$. All the result are 0 in $\mathbb{F}_2$. Hence C is self-orthogonal.

Theorem 2.1.20. When C be an $[n,k]$-linear code, $dim(C) + dim(C^\perp) = n$.

Proof. Follows immediately by Theorem 2.1.16. □

Proposition 2.1.21. Assume that a code C is an $[n,k]$-linear code which is self-dual, then n is even.

Proof. Suppose C is an $[n,k]$-linear code over $\mathbb{F}_q$. By Theorem 2.1.20, $C^{\perp}$ is of dimension $n-k$. Because the code C is self-dual, dimension of the code and its dual equal, $k=n-k$. Then $n=2k$. □

Now, we present the Hamming distance and then the Hamming weight. They are the most commonly used notations for codes over finite fields. They are defined as follows.

Definition 2.1.22. The Hamming distance between two vectors (words) $\bar{u},\bar{v}\in\mathbb{F}_{\mathrm{q}}^{\mathrm{n}}$, $d_H(\bar{u},\bar{v})$, is defined by the number of different coordinate positions in $\bar{u}=(u_1,u_2,\dots,u_n)$ and $\bar{v}=(v_1,v_2,\dots,v_n)$, that is,

$$d_H(\bar{u},\bar{v}) \coloneqq |\{i|1\leq i\leq n, u_i\neq v_i\}| \tag{2.4}$$

Definition 2.1.23. The Hamming weight of a vector (word) $\bar{u}$, $w_H(\bar{u})$, is the number of nonzero coordinates of $\bar{u}$, that is,

$$w_H(\bar{u}) \coloneqq |\{i|1\leq i\leq n, u_i\neq 0\}| \tag{2.5}$$

Example 2.1.24. Assume that the words $\bar{u}=10101101$ and $\bar{v}=01011100$ are in $\mathbb{F}_2^8$. Then,

$$d_H(\bar{u},\bar{v})=d(10101101,01011100)=5.$$

The Hamming weights of $\bar{u}$ and $\bar{v}$ are $w_H(\bar{u})=w_H(10101101)=5$, $w_H(\bar{v})=w_H(01011100)=4$.

Definition 2.1.25. *The minimum distance of a code C* is defined by

$$d(C)\coloneqq \min\{d_H(\bar{u},\bar{v})\,|\bar{u},\bar{v}\in C,\bar{u}\neq\bar{v}\} \tag{2.6}$$

Definition 2.1.26. *The minimum weight of a code C* is defined as

$$w(C)\coloneqq \min\{w_H(\bar{u})|\bar{u}\neq\bar{0},\bar{u}\in C\} \tag{2.7}$$

Assume C be a code over $\mathbb{F}_q$. If $|C|=M$ and C is of length n and the minimum distance d then it is called an (n,M,d)-code. If this code is linear with dimension k, then it is written as $[n,k,d]$-code. Therefore the values of n, M, k, d are named the parameters of a code.

Theorem 2.1.27. Assume that C is an $[n,k,d]$-linear code over $\mathbb{F}_q$, therefore

$$d(C) = w(C).$$

Proof: Let C be a linear code with minimum weight $w(C)$ and minimum distance $d(C)$. Then there exist nonzero codewords $\bar{a}, \bar{b}, \bar{c} \in C$ such that $d_H(\bar{a}, \bar{b}) = d(C)$ and $w_H(\bar{c}) = w(C)$. Here, $d_H(\bar{a}, \bar{b}) = w_H(\bar{a} - \bar{b})$ since whenever a coordinate position of codewords $\bar{a}$ and $\bar{b}$ differ, the corresponding coordinate position of $\bar{a} - \bar{b}$ will not be 0. Thus, we have

$$d(C) \leq d_H(\bar{c}, \bar{0}) = w_H(\bar{c} - \bar{0}) = w_H(\bar{c}) = w(C) \leq w_H(\bar{a} - \bar{b}) = d_H(\bar{a}, \bar{b}) = d(C).$$

So, $d(C) = w(C)$. □

Example 2.1.28. Assume that the code C is a binary linear code whose generating matrix is $\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$. We will find the parameters of C.

We get $C = \{000000, 100110, 010101, 001011, 110011, 101101, 011110, 111000\}$. The Hamming weights of the codewords of C are;

$$w_H(000000) = 0, \quad w_H(100110) = 3, \quad w_H(010101) = 3, \quad w_H(001011) = 3,$$
$$w_H(110011) = 4, \quad w_H(101101) = 4, \quad w_H(011110) = 4, \quad w_H(111000) = 3.$$

Thus, $w(C) = 3$. Since the code C is a linear code, $d(C) = w(C)$. Then this code is a $[6, 3, 3]$-code.

Definition 2.1.29. Suppose that C is a binary self dual code. When some codewords in C have weigth not divisible by 4 then it is said to be a *Type I (singly even)* code. When all codewords in C have weights divisible by 4, C is said to be a *Type II (doubly even)* code.

It is shown in [14] that for a doubly even self dual code of length n with the minimum weight d, we have

$$d \leq 4\left\lfloor\frac{n}{24}\right\rfloor + 4 \qquad (2.8)$$

Definition 2.1.30. For a doubly even self-dual code, if the minimum distance d attains the bound in inequality (2.8), then this code is called *extremal*.

Now, we will give the definitions of a ball and sphere.

Definition 2.1.31. Let $\bar{x} \in \mathbb{F}_q^n$, $e \in \mathbb{N}$. Then a *ball* with radius e and center $\bar{x}$ is the set such that

$$B_e(\bar{x}) \coloneqq \{\bar{y} \in \mathbb{F}_q^n \big| \, d_H(\bar{x}, \bar{y}) \leq e\} \tag{2.9}$$

The following is called a *sphere*

$$S_e(\bar{x}) \coloneqq \{\bar{y} \in \mathbb{F}_q^n \big| \, d_H(\bar{x}, \bar{y}) = e\} \tag{2.10}$$

Lemma 2.1.32. Assume that C is a code (it can be nonlinear) of length n over $\mathbb{F}_q$ with minimum distance $d = 2e + 1$. Also suppose that $\bar{c}_1, \bar{c}_2, \dots, \bar{c}_m$ are the codewords of the code C. Then $B_e(\bar{c}_i) \cap B_e(\bar{c}_j) = \emptyset$ for all $i \neq j$.

Proof. Let $\bar{a} \in \mathbb{F}_q^n$ and $\bar{a} \in B_e(\bar{c}_i) \cap B_e(\bar{c}_j)$. Then by the definition of the ball $d_H(\bar{c}_i, \bar{a}) \leq e$ and $d_H(\bar{c}_j, \bar{a}) \leq e$. Now, we give the proof of the triangle inequality. Let $\bar{a} = a_1 a_2 \dots a_n$, $\bar{b} = b_1 b_2 \dots b_n$ and $\bar{x} = x_1 x_2 \dots x_n$. Now $d_H(\bar{a}, \bar{x})$ is the number of values i for $a_i \neq x_i$. If $a_i \neq x_i$, then either $a_i \neq b_i$ or $b_i \neq x_i$. So,

$$\{i \mid a_i \neq x_i\} \subseteq \{i \mid a_i \neq b_i\} \cup \{i \mid b_i \neq x_i\}.$$

$$|\{i \mid a_i \neq x_i\}| \leq |\{i \mid a_i \neq b_i\} \cup \{i \mid b_i \neq x_i\}|,$$

$$\leq |\{i \mid a_i \neq b_i\}| + |\{i \mid b_i \neq x_i\}|,$$

$$d_H(\bar{a}, \bar{x}) \leq d_H(\bar{a}, \bar{b}) + d_H(\bar{b}, \bar{x}).$$

Hence, $d_H(\bar{c}_i, \bar{c}_j) \leq d_H(\bar{c}_i, \bar{x}) + d_H(\bar{x}, \bar{c}_j) \leq 2e$. This is a contradiction since $d = 2e + 1$. □

Now, we consider the value of e in the definition of the ball as the number of changed coordinate positions of original codeword $\bar{x}$. Later, we will call this number e errors in $\bar{x}$. Consider a code C with minimum distance at least $2e + 1$. If we take any codeword $\bar{c} \in C$ and change t coordinates, where $t \leq e$ (that is we make t errors on the codeword $\bar{c}$), then by previous lemma it can be taken as original codeword $\bar{c}$, it means we can correct t-errors. This implies that the balls with radius e around different codewords are disjoint. That means if we change e coordinates of different codewords, then all words getting are different.

Theorem 2.1.33. If a code C is over $\mathbb{F}_q$ of length n with $d = 2e + 1$, so

$$|C| \cdot \sum_{t=0}^{e} \binom{n}{t} (q-1)^t \leq q^n. \tag{2.11}$$

Proof. Here, q^n is the number of all possible words in $\mathbb{F}_q^n$. The left hand side of (2.11) is the number of vectors in the balls of radius e. For each t there exists $\binom{n}{t}$ different positions which can be changed, and there exists $q-1$ different possible value for each changed positions. Now, we will demonstrate the number of words in balls of radius e, $|B_e(\overline{c_\iota})|$, $\overline{c_\iota} \in C$. Then,

$$|B_e(\overline{c_\iota})| = \left|\{\bar{x} \in \mathbb{F}_q^n |\ d_H(\bar{x}, \overline{c_\iota}) \leq e\}\right| = \left|\bigcup_{r=0}^{e} S_r(\overline{c_\iota})\right| \qquad \text{for all } i \in \{1, \dots, n\} \tag{2.12}$$

By the definition of the sphere if $r \neq s$, then $S_r(\overline{c_\iota}) \cap S_s(\overline{c_\iota}) = \emptyset$. Thus, the following equality

$$|B_e(\overline{c_\iota})| = \sum_{r=0}^{e} |S_r(\overline{c_\iota})|$$

is obtained. Here, $|S_0(\overline{c_\iota})| = 1$, since there is no change on the codeword $\overline{c_\iota}$. $|S_1(\overline{c_\iota})| = \binom{n}{1}(q-1)$, since 1 coordinate will be changed and we have $\binom{n}{1}$ different possibilities for this, $(q-1)$ different values can be filled in this coordinate. $|S_2(\overline{c_\iota})| = \binom{n}{2}(q-1)^2$, since 2 coordinate will be changed and we have $\binom{n}{2}$ different possibilities for this and $(q-1)^2$ different values can be filled in these coordinates. So in general, $|S_r(\overline{c_\iota})| = \binom{n}{r}(q-1)^r$. Hence,

$$|B_e(\overline{c_\iota})| = \sum_{r=0}^{e} |S_r(\overline{c_\iota})| = \sum_{r=0}^{e} \binom{n}{r} (q-1)^r \qquad \text{for all } \overline{c_\iota} \in C.$$

By Lemma 2.1.32, we have $B_e(\overline{c_\iota}) \cap B_e(\overline{c_J}) = \emptyset$ for any distinct codewords $\overline{c_\iota}, \overline{c_J} \in C$. So, $\bigcup_{i=1}^{M} B_e(\overline{c_\iota}) \subseteq \mathbb{F}_q^n$, where M is the cardinality of C, $|C|$. Then we get

$$|C| \cdot \sum_{t=0}^{e} \binom{n}{t} (q-1)^t \leq q^n. \qquad \square$$

The bound above is called as the *Hamming bound* or the *sphere packing bound*. When equality holds, we call C as a *perfect code*. This means that every word in $\mathbb{F}_q^n$ is covered by precisely one sphere of radius e around a codeword.

Golay codes are important examples of perfect codes in coding theory. More information about Golay codes can be found in [15].

Example 2.1.34. Consider the $[23,12,7]$ binary Golay code. Its length is 23, dimension is 12, and $q = 2$, $|C| = 2^{12}$, $d = 7 = 2e + 1$, $e = 3$. Then, the following equality

$$2^{12} \cdot \sum_{t=0}^{3} \binom{23}{t} (2-1)^t = 2^{23}$$

is obtained. Since this equality satisfies the equation for perfect codes, then $[23,12,7]$-code is perfect.

Definition 2.1.35. Suppose $t \in \mathbb{N}$ and C is a code over some alphabet A. The code C is *t-error detecting code* if C can detect all errors up to t, but cannot detect at least one $t+1$ errors.

Theorem 2.1.36. Suppose C is a code with minimum distance d, then C is $(d-1)$-error detecting code.

Proof. Suppose $d(C) = d$. Let $\bar{c} \in C$ and $\bar{x}$ be a received word with at most $d-1$ errors on $\bar{c}$. Received words must be in the same code but we have, $1 \leq d_H(\bar{x}, \bar{c}) \leq d - 1 < d$, then $\bar{x}$ is not in C; hence, this code detects $d-1$ errors. Assume $\bar{u}$ and $\bar{v}$ are two codewords of C such that $d_H(\bar{u}, \bar{v}) = d$ and $\bar{u}$ is a received word at least d errors. Then $\bar{u}$ can be received as $\bar{v}$ in a transmission. Since both $\bar{u}, \bar{v}$ are in C, then C cannot detect at least d errors. Therefore, C is a $d-1$ error detecting code. $\square$

Definition 2.1.37. Assume that C is a code over some alphabet A. Then, C is called a *v-error correcting code* if C can correct every error up to v but cannot correct at least one $v+1$ errors.

Lemma 2.1.38. A code of minimum distance $2v + 2$ cannot correct $v + 1$ errors.

Proof. Suppose $\bar{x}$ and $\bar{y}$ are in the code C such that $d_H(\bar{x}, \bar{y}) = 2v + 2$. Without loss of generality, assume $\bar{x} = x_1 x_2 \dots x_n$, $\bar{y} = y_1 y_2 \dots y_n$, $x_1 \neq y_1$, $x_2 \neq y_2$, $\dots$, $x_{2v+2} \neq y_{2v+2}$, $x_{2v+3} = y_{2v+3}$, $\dots$, $x_n = y_n$, take $\bar{x}$ as $\bar{x} = x_1 x_2 \dots x_{2v+2}\, y_{2v+3} y_{2v+4} \dots y_n$. Let us define $\overline{x_e}$ which is $\bar{x}$ with $v + 1$ errors at the first $v + 1$ coordinates such as $\overline{x_e} = y_1 y_2 \dots y_{v+1} x_{v+2} x_{v+3} \dots x_{2v+2} y_{2v+3} y_{2v+4} \dots y_n$. Then,

$d(\overline{x_e}, \bar{x}) = v + 1$, $d(\overline{x_e}, \bar{y}) = v + 1$. So, $\overline{x_e} \in B_{(v+1)}(\bar{x})$ and $\overline{x_e} \in B_{(v+1)}(\bar{y})$.

Hence $B_{(v+1)}(\bar{x}) \cap B_{(v+1)}(\bar{y}) \neq \emptyset$. Since this intersection is not an empty set, one can take $\overline{x_e}$ as $\bar{x}$ or $\bar{y}$. Hence errors cannot be corrected. □

Theorem 2.1.39. If C is a code with $d(C) = d$, then C is a $\left\lfloor \frac{d-1}{2} \right\rfloor$-error correcting code.

Proof. Firstly, let $d = 2e + 1$ be odd. Let $\bar{r}, \bar{s} \in C$ and $\bar{x}$ is a word in at most e-errors on $\bar{r}$, $d_H(\bar{r}, \bar{x}) \leq e$. Then,

$$d_H(\bar{r}, \bar{s}) \leq d_H(\bar{x}, \bar{r}) + d_H(\bar{x}, \bar{s}),$$
$$d_H(\bar{r}, \bar{s}) - d_H(\bar{x}, \bar{r}) \leq d_H(\bar{x}, \bar{s}),$$
$$2e + 1 - e \leq d_H(\bar{x}, \bar{s}),$$
$$e + 1 \leq d_H(\bar{x}, \bar{s}).$$

Thus, $\bar{x}$ will be taken as $\bar{r}$ since the Hamming distance of $\bar{x}$ and $\bar{r}$ is less than the Hamming distance of $\bar{x}$ and $\bar{s}$.

Let $\bar{r}$, $\bar{s} \in C$ such that $d_H(\bar{r}, \bar{s}) = 2e + 1$ and without loss of generality, $\bar{r} = r_1 r_2 \dots r_n$, $\bar{s} = s_1 s_2 \dots s_n$ with $r_1 \neq s_1$, $r_2 \neq s_2$, $\dots$, $r_{2e+1} \neq s_{2e+1}$, $r_{2e+2} = s_{2e+2}$, $\dots$, $r_n = s_n$, and take $\bar{r} = r_1 r_2 \dots r_{2e+1} s_{2e+2} \dots s_n$. Let us define $\overline{x_e}$ which is $\bar{x}$ with $e + 1$ errors at the first $e + 1$ coordinates such as $\overline{x_e} = s_1 s_2 \dots s_{e+1} r_{e+2} \dots r_{2e+1} s_{2e+2} \dots s_n$. Then, $d(\overline{x_e}, \bar{r}) = e + 1$, $d(\overline{x_e}, \bar{s}) = e$. Then $\overline{x_e}$ will be taken as $\bar{s}$ since the Hamming distance between these words is minimal, which is wrong.

For the second, let $d = 2v + 2$ be even. Let $\overline{u_1}, \overline{u_2} \in C$ and $\bar{x}$ is a word in at most e-errors on $\overline{u_1}$, $d_H(\overline{u_1}, \bar{x}) \leq e$. Then,

$$d_H(\overline{u_1},\overline{u_2}) \leq \ d_H(\bar{x},\overline{u_1}) + d_H(\bar{x},\overline{u_2}),$$
$$d_H(\overline{u_1},\overline{u_2}) - d_H(\bar{x},\overline{u_1}) \leq d_H(\bar{x},\overline{u_2}),$$
$$2e + 2 - e \leq d_H(\bar{x},\overline{u_2}),$$
$$e + 2 \leq d_H(\bar{x},\overline{u_2}).$$

Hence, $\bar{x}$ will be taken as $\overline{u_1}$, correctly. If $e+1$ errors occur, the code cannot correct, we proved this case in the Lemma 2.1.38. Thus, we proved the theorem. □

Example 2.1.40. Suppose C is a code over $\mathbb{F}_3$ such that $C = \{000000, 000111, 111222\}$. Then, we will find the error detection and correction capabilities of this code.

$$d(C) = \min\{d_H(\overline{c_\imath},\overline{c_\jmath}) \mid \overline{c_\imath},\overline{c_\jmath} \in C, i \neq j\} = 3,$$

Error detection capability of C is $3-1=2$, error correction capability of C is $\left\lfloor\frac{3-1}{2}\right\rfloor = 1$.

Definition 2.1.41. Assume that $r > 2$ and $n = 2^r - 1$. Consider $\text{Ham}(r, 2)$ to be a *binary Hamming* code of length $2^r - 1$ given by the parity check matrix H whose columns consist of all nonzero vectors of $\mathbb{F}_2^r$.

Example 2.1.42. We will give the parity check matrix, H, for $\text{Ham}(3,2)$ and its all codewords. So, we have

$$H = \begin{bmatrix} 1\ 0\ 1\ 1\ 1\ 0\ 0 \\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ 1\ 1\ 1\ 0\ 0\ 0\ 1 \end{bmatrix}.$$

By Theorem 2.1.11, we get the generating matrix for $\text{Ham}(3,2)$ where $H = [-A^T|I]$, A is a 4×3 matrix, as

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

Then all codewords of $\text{Ham}(3,2)$ are in the following table.

Table 2.1: Codewords of $\mathrm{Ham}(3,2)$.

0000000	0001110	0101101	1110001
1000111	1100100	0011011	1101010
0100011	1010010	1100011	1011100
0010101	1001001	0011011	1111111

Theorem 2.1.43. For a linear code C, consider H to be a parity check matrix. Then $d(C) = d$ if and only if every subsets of columns of H of size $d-1$ is linearly independent and there exists a linearly dependent subset of columns of size d.

Proof. Let C and H be taken as mentioned and $\bar{s} = s_1 s_2 \ldots s_n \in \mathbb{F}_q^n$. By Definition 2.1.10, when $\bar{s}$ is a codeword, $H\bar{s}^T = \bar{0}$. Suppose $H = [H_1|H_2|\ldots|H_n]$, then $H_1 s_1 + H_2 s_2 + \cdots + H_n s_n = \bar{0}$. Assume $w_H(\bar{s}) = d$, then $\exists i_1 i_2 \ldots i_d \in \{1,2,\ldots,n\}$ such that $s_{i_j} \neq 0$ for all $j \in \{1,2,\ldots,d\}$, and $s_{i_k} = 0$ for $k \notin \{1,2,\ldots,d\}$. In this case we obtain, $s_{i_1} H_{i_1} + s_{i_2} H_{i_2} + \cdots + s_{i_d} H_{i_d} = \bar{0}$. So, $\{H_{i_1}, H_{i_2}, \ldots, H_{i_d}\}$ are linearly dependent. So, for the linearly dependent set d is the minimum number of columns of H. □

Theorem 2.1.44. $d(\mathrm{Ham}(r,2)) = 3$ for all $r > 2$.

Proof. If $\bar{s}, \bar{t} \in \mathbb{F}_2^n$, $\bar{s} \neq \bar{0}, \bar{t} \neq \bar{0}$, then $\{\bar{s}, \bar{t}\}$ linearly dependent if and only is $\bar{s} + \bar{t} = \bar{0}$. Since we are in $\mathbb{F}_2$, $\bar{s} = \bar{t}$. Thus any two distinct nonzero vectors are linearly independent. So, any two columns of $\mathrm{Ham}(r,2)$ are linearly independent and some of three columns of it is linearly dependent. □

Binary Hamming codes are the family of perfect codes. Since H is an $r \times (2^r - 1)$ matrix, then $k = 2^r - 1 - r$. Its minimum distance $d = 3$ hence it is exactly single error correcting of parameters $[2^r - 1, 2^r - r - 1, 3]$.

Definition 2.1.45. If $C = \{\alpha\alpha \ldots \alpha | \alpha \in \mathbb{F}_2\}$ such a code, we call this code as the *binary repetition code* of length n and denoted by $[n, 1, n]$-code.

Binary repetition codes correct the errors up to $e = \left\lfloor \frac{n-1}{2} \right\rfloor$. A generating matrix for these codes is

$$G = [1|11 \dots 1].$$

The corresponding parity check matrix is

$$H = \left[\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} I_{n-1} \right].$$

Definition 2.1.46. *The information rate,*R*, of* a code C of length n over $\mathbb{F}_q$ is defined as $R := \frac{\log_q |C|}{n}$.

For the binary repetition code, the information rate is $\frac{1}{n}$. That means for one bit information, we have to send n bits of symbols.

Example 2.1.47. Let 0101 be the information that we want to send and we replace each 0 with 00000 and each 1 with 11111. Then, 0101 will be replaced by 00000111110000011111. Here the information rate is $\frac{1}{5}$.

Definition 2.1.48. Consider C be any code over $\mathbb{F}_q$ of length n. Then *the extended code* of this code, $\bar{C}$, is defined as follows

$$\bar{C} := \left\{ \left(c_1, c_2, \dots, c_n, -\sum_{i=1}^{n} c_i \right) : (c_1, c_2, \dots, c_n) \in C \right\} \tag{2.13}$$

If $q = 2$, the sum $-\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} c_i$ is named as the parity-check coordinate.

Example 2.1.49. Assume that we have the binary code $C = \{0000, 1010, 0101, 1001\}$. The extended code of C is $\bar{\mathrm{C}} = \{00000, 10100, 01010, 10010\}$.

Regarding the parameters n, M, d, we have the following properties for the (n, M, d) codes.

1. if d increases, the code detects and corrects more errors,

2. if the information rate increases, the code transports the message faster since the number of sending symbols decreases.

To increase information rate, n should decrease and M should increase. However, if M increases then d decreases. The value of d is found by the all combinations of the codewords, and if M increases there will be more codewords in that combinations, then d decreases. So, these values are contrary for finding an optimum error correcting code which is important subject in coding theory. We call these optimum error correcting codes as *good codes* which are with high information rate and high error correction capacity.

Repetition codes have least information rate, $\frac{1}{n}$, and least error correcting capacity, $\left\lfloor\frac{n-1}{2}\right\rfloor$, then they are not good codes.

The Hamming code of length 7 has the best value of the information rate, $\frac{4}{7}$, for one error correcting. For each 4 bits of information, the code sends 7 bits as in the following rule.

Now, we will give the rule for Hamming code of length 7 that the code sends 4 bits of message as 7 bits. Let $r_1r_2r_3r_4$ be a sending message. Since the Hamming code is over $\mathbb{F}_2$ then parity bits of this are given as $p_1 = r_1+r_3 + r_4$, $p_2 = r_1 + r_2 + r_4$ and $p_3 = r_1+r_2+r_3$. Therefore, for $r_1 = 1$ and $r_2 = r_3 = r_4 = 0$, then $p_1 = 1$, $p_2 = 1$ and $p_3 = 1$. For $r_2 = 1$ and $r_1 = r_3 = r_4 = 0$, then $p_1 = 0$, $p_2 = 1$ and $p_3 = 1$. For $r_3 = 1$ and $r_1 = r_2 = r_4 = 0$, then $p_1 = 1$, $p_2 = 0$ and $p_3 = 1$. For $r_4 = 1$ and $r_1 = r_2 = r_3 = 0$, then $p_1 = 1$, $p_2 = 1$ and $p_3 = 0$. Then the columns of the generating matrix for this code is the coefficients of r_1, r_2, r_3, r_4 in r_1, r_2, r_3, r_4, p_1, p_2 and p_3. Then we have the following generating matrix

$$G = \begin{bmatrix} 1000111 \\ 0100011 \\ 0010101 \\ 0001110 \end{bmatrix}.$$

Definition 2.1.50. *The weight enumarator* of C, where it is a code of length n over $\mathbb{F}_q$, is

$$W_C(z) = \sum_{\bar{c}\in C} z^{w_H(\bar{c})} = \sum_{i=0}^{n} A_i z^i \tag{2.14}$$

where A_i indicates that the number of codewords of Hamming weight i in this code.

Example 2.1.51. Let us consider the code C in Example 2.1.28. This code has 1 codeword of Hamming weight 0, 4 codewords of Hamming weight 3, 3 codewords of Hamming weight 4. Thus,

$$W_C(z) = 1 + 4z^3 + 3z^4.$$

2.2. BACKGROUNDS ON COMBINATORIAL DESIGN THEORY

We give a short overview of some of the basic notations and concepts of combinatorial design theory. The proofs of the results which are presented in this chapter and more details about combinatorial design theory can be found in [15] and [16].

Definition 2.2.1. Consider P as a set of v points and $\mathcal{B}$ as a family of k-subsets of P, called blocks. However, A t-(v, k, λ) design is a pair $(P, \mathcal{B})$ such that each distinct t-subset of P occurs in precisely λ blocks.

There are two different notations widely used for a design, those are t-(v, k, λ) and $S_\lambda(t, k, v)$. When $\lambda = 1$, that is, a t-$(v, k, 1)$ design is denoted by $S(t, k, v)$.

Example 2.2.2. For $P = \{0, 1, 2, 3\}$, the blocks $\mathcal{B} = \big\{\{0,1,2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}\big\}$ form a 2-$(4, 3, 2)$ design. P has 4 points, so $v = 4$, each block contains 3 points, so $k = 3$, we look at each pair of points, $t = 2$, each pair is in 2 blocks, so $\lambda = 2$.

Example 2.2.3. Let us consider a design by taking all nonzero codewords of $\mathbb{F}_2^4$ and we take blocks as triples $\{\bar{a}, \bar{b}, \bar{c}\}$ with $\bar{a} + \bar{b} + \bar{c} = \bar{0}$, where $\bar{a}$, $\bar{b}$ and $\bar{c}$ are all different. Here, $\bar{a} + \bar{b} = \bar{c}$ and any pair $\bar{a}$ and $\bar{b}$ with $\bar{a} \neq \bar{b}$ assigns the value of $\bar{c}$, different from $\bar{a}$ and $\bar{b}$. Thus, $t = 2$ and $\lambda = 1$. The number of points is $2^4 - 1 = 15$, so $v = 15$, blocks consist of 3 points, so $k = 3$. So we have an $S(2, 3, 15)$ and we also denote this design 2-$(15, 3, 1)$.

Definition 2.2.4. The number r is *the replication number* of a design that for a fixed point $p \in P$, there are r blocks containing p.

Example 2.2.5. Consider $(P, \mathcal{B})$ be a 2-$(7, 3, 1)$ design, where $P = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathcal{B} = \{\{1, 2, 5\}, \{1, 3, 6\}, \{1, 4, 7\}, \{2, 3, 7\}, \{2, 4, 6\}, \{3, 4, 5\}, \{5, 6, 7\}\}$. In this design, there are 7 points and each block has 3 points, so we have $v = 7$, $k = 3$. Observe that each two points of

P is in exactly one block, so we have $\lambda = 1$. Each point appears 3 times in all blocks, so the replication number r is 3.

The following theorem, describes the cardinality of $\mathcal{B}$ of a t-$(v\ ,k,\lambda)$ design.

Theorem 2.2.6. A t-(v,k,λ) design has exactly

$$b = \lambda \binom{v}{t} \Big/ \binom{k}{t}$$

blocks.

Proof: Let $(P,\mathcal{B})$ be a t-(v,k,λ) design, and $b = |\mathcal{B}|$. Let us define a set

$$I = \{(\{p_1,p_2,\dots,p_t\},B) \mid p_1,p_2,\dots,p_t \in P, B \in \mathcal{B}, \{p_1,p_2,\dots,p_t\} \subseteq B\}.$$

Here, we consider the cardinality of I in two different ways. We know that the number of the subsets of P with t points is $\binom{v}{t}$. By the definition of a design these subsets belong exactly λ blocks. Thereby $|I|$ is $\lambda\binom{v}{t}$. On the other hand the number of subsets of blocks with t points is $\binom{k}{t}$ and there are b blocks, so $|I|$ is $b\binom{k}{t}$. Hence, $b\binom{k}{t} = \lambda\binom{v}{t}$. □

Theorem 2.2.7. Every t-(v,k,λ) design is also an i-(v,k,λ_i) design for $0 \leq i \leq t$, where

$$\lambda_i = \lambda \binom{v-i}{t-i} \Big/ \binom{k-i}{t-i}.$$

Proof. Let $(P,\mathcal{B})$ be a t-(v,k,λ) design and let λ_i denote the number of blocks consisting $\{p_1,p_2,\dots,p_i\} \subseteq P$. Let u define a set

$$I = \{(\{p_{i+1},p_{i+2},\dots,p_t\},B) \mid p_1,p_2,\dots,p_t \in P, B \in \mathcal{B}, \{p_1,p_2,\dots,p_i,\dots,p_t\} \subseteq B\}.$$

We count the cardinality I in two different ways. We know that there are $\binom{v-i}{t-i}$ t-subsets which contain $\{p_1,p_2,\dots,p_i\}$. By the definition of a design, each t-subset occurs in λ blocks, so we have the number of t-subsets which contains $\{p_1,p_2,\dots,p_i\}$ is $\lambda\binom{v-i}{t-i}$. The number of blocks that contain $\{p_1,p_2,\dots,p_i\}$ is $\binom{k-i}{t-i}$ and there are λ_i blocks containing this set. Then the number of t-subsets which contain $\{p_1,p_2,\dots,p_i\}$ is $\lambda_i\binom{k-i}{t-i}$. Hence $\lambda_i = \lambda\binom{v-i}{t-i}/\binom{k-i}{t-i}$. Besides that, since each distinct i-subset is in exactly λ_i blocks, then this t-design is also i-(v, k, λ_i) with $\lambda = \lambda_i$. □

The number λ_1 specified in the proof of Theorem 2.2.7. is usually denoted by r, replication number. The set $\{p_1, p_2, \dots, p_i\}$ can be thought for each element of P, it means $i = 1$. We give two particular cases of Theorem 2.2.7. which are

$$bk = vr,$$

$$\lambda(v-1) = r(k-1).$$

For the first equality, we count the number of appearances of all points of P. Since there are b blocks and each block has k points, then the appearance of each point of P is bk. Since there are v points and each point appears r times, then the appearance of each point of P is vr.

For the second equality, we have $\frac{bk(k-1)}{2} = \frac{\lambda v(v-1)}{2}$ since $b\binom{k}{t} = \lambda\binom{v}{t}$ with $t = 2$. Then we get $\lambda(v-1) = r(k-1)$ since $bk = vr$.

Definition 2.2.8. A t-(v, k, λ) design is called simple when it does not contain any repeated blocks, where if two blocks in a design are exactly same.

Example 2.2.9. Consider the point set as $P = \{1,2,3,4,5,6\}$, and the block set as $\mathcal{B} = \big\{\{1,2\},\{2,3\},\{3,4\},\{1,4\},\{5,6\},\{5,6\}\big\}$. Here, $v = 6$, $k = 2$, $t = 1$, $\lambda = 2$, so we have 1-$(6, 2, 2)$ design with repeated blocks.

Observe that, if $\lambda = 1$, there does not exist any repeated blocks.

Designs can be presented algebraically by incidence matrices. Now, we give the definition of the incidence matrix of a design.

Definition 2.2.10. Consider $D = (P, \mathcal{B})$ be a t-design (the term is used instead of any t-(v, k, λ) design). Consider the $P = \{x_1, x_2, \dots, x_v\}$ and $\mathcal{B} = \{B_1, B_2, \dots, B_b\}$. Therefore, the $(b \times v)$-matrix $A = (a_{ij})$, where $1 \le i \le b$, and $1 \le j \le v$, is defined by

$$a_{ij} := \begin{cases} 1, if\ x_j \in B_i, \\ 0, if\ x_j \notin B_i \end{cases}$$

and it is called the *incidence matrix of D*.

Observe that, the incidence matrix A of a t-(v,k,λ)-design provides the followings:

1. Each column of A has exactly r 1's.
2. Each row of A has exactly k 1's.
3. Two distinct rows of A both have 1's in exactly λ columns.

Example 2.2.11. Consider the 2-$(7,3,1)$ design represented in Example 2.2.5. Let us determine the blocks containing point 1 and point 4 as an example. The point 1 is in the first, second and third blocks, so we have $a_{11}=1$, $a_{21}=1$, $a_{31}=1$, $a_{41}=0$, $a_{51}=0$, $a_{61}=0$, $a_{71}=0$. The point 4 is in first, third and fourth blocks, so we have $a_{14}=0$, $a_{24}=0$, $a_{34}=1$, $a_{44}=0$, $a_{54}=1$, $a_{64}=1$, $a_{74}=0$. Hence, this design has the following (7×7) incidence matrix:

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

Observe that, rows of A indicate that the blocks of B and columns of A indicate that the appearance of the points of P where a_{ij} is 1 if and only if the jth point is in the ith block.

A 2-(v,k,λ) is called symmetric design if $v=b$ and $k=r$. In these symmetric designs, each pair of blocks in the block set have λ points in common. Observe that, symmetric designs have an incidence matrix which is square.

Example 2.2.12. Let us consider the 2-$(7,4,2)$ symmetric design with the point and block sets as $P=\{1,2,3,4,5,6,7\}$ and $B=\{\{1,2,3,7\},\{1,2,4,6\},\{1,3,4,5\},\{1,5,6,7\},\{3,4,6,7\},\{2,3,5,6\},\{2,4,5,7\}\}$.

Therefore, its incidence matrix is given in the following.

$$A = \begin{bmatrix} 1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,0\,1\,0\,1\,0 \\ 1\,0\,1\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1\,1\,1 \\ 0\,0\,1\,1\,0\,1\,1 \\ 0\,1\,1\,0\,1\,1\,0 \\ 0\,1\,0\,1\,1\,0\,1 \end{bmatrix}.$$

Observe that, every row and every column has four 1's, which is equal the value of k and r, respectively. So, $k = r$. Therefore, each pair of rows have 2 common points, that is $\lambda = 2$, and each two points are exactly in two blocks, that is $t = 2$.

Observe that, when $k = t$, t-designs always exist if the blocks of the designs is t-subsets of the point set. Also, when $k = v$, t-designs always exist since all points are in one block. Then such t-designs are called *trivial designs*.

2.3. THE CONNECTIONS BETWEEN CODES AND COMBINATORIAL DESIGNS

We begin with the definition of support which is important to present the relations between codes and combinatorial designs.

Definition 2.3.1. For a codeword $\bar{x} \in \mathbb{F}_q^n$, such that $\bar{x} = x_1 x_2 \dots x_n$, the following set

$$supp(\bar{x}) \coloneqq \{i | \; x_i \neq 0\}$$

is called the *support* of $\bar{x}$ that consists of all coordinate position of non-zero components.

Example 2.3.2. Let $C = \{c_1, c_2, c_3\}$ where $c_1 = 201102$, $c_2 = 110022$, $c_3 = 101011$ be ternary code. Then, supports of the codewords c_1, c_2 and c_3 are $\{1, 3, 4, 6\}$, $\{1, 2, 5, 6\}$ and $\{1, 3, 5, 6\}$, respectively.

The main problem here is to obtain designs from codes. Obtaining designs from codes is a more complex problem than obtaining codes from designs. Given any design, its incidence matrix can be taken and a code can be constructed by taking this incidence matrix as a generating matrix or even a parity-check matrix for this code. The difficulty in constructing designs from codes is that designs require special conditions such as all t-subsets of the point set must be in exactly λ blocks and the number of these blocks is bound by the values of t, v and k.

By [17], the next theorem gives a relationship between t-designs and perfect codes.

Theorem 2.3.3. Assume that C is a perfect code of length n over $\mathbb{F}_q$ with the minimum distance $d = 2t + 1$. Then, C is a $(t+1)$-$(n, d, (q-1)^t)$ design.

Proof. Let T is a subset with $(t+1)$ elements, of the set $\{1, 2, \dots, n\}$, as a support of any word $\overline{w} \in \mathbb{F}_q^n$ of weight $(t+1)$. Then this set can be filled with nonzero coordinate values in $(q-1)^{t+1}$ ways. Then, each such $\overline{w}$ must be at distance t or less from exactly one codeword $\bar{c} \in C$. This codeword must have weight d, and it must be same with all of the nonzero coordinates of $\overline{w}$. By this way, the codewords are all distinct and T is contained in exactly $(q-1)^t$ codewords of weight $2t+1$. Hence, the set $\{1, 2, \dots, n\}$ is our point set, that is $v = n$, the block set is all codewords of weight d, that is $k = d$, each $(t+1)$ elements from the set $\{1, 2, \dots, n\}$ is in exactly $(q-1)^t$ codewords having weight $2t+1$, that is $\lambda = 2t + 1$. □

The following example is about Theorem 2.3.3 with the $[7, 4, 3]$-Hamming code.

Example 2.3.4. Consider C be a perfect $[7, 4, 3]$ Hamming code over $\mathbb{F}_2$ with $n = 7$, $d = 3$, $t = 1$ since $d = 2t + 1$, and $q = 2$. The word $\overline{w} \in \mathbb{F}_2^7$ has weight 2 since $t + 1 = 2$. Then we have a 2-$(7, 3, 1)$ design. It can be choosen the point set as $P = \{1, 2, 3, 4, 5, 6, 7\}$ and block set as $B = \{\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 7\}, \{1, 5, 6\}, \{2, 6, 7\}, \{1, 3, 7\}\}$.

It can be seen that any two points of P, that is $t + 1 = 2$, are found in just one block, that is $\lambda = 1$. This gives a 2-$(7, 3, 1)$ design and it is known as Fano Plane.

Now, we give in the following diagram the usual representation of Fano Plane.

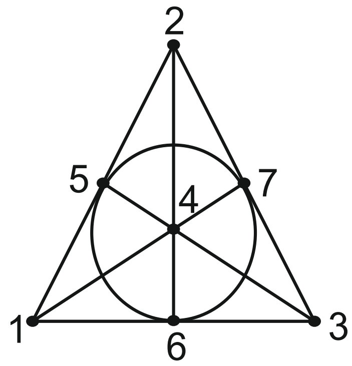

Figure 2.1: Fano Plane.

We take blocks of the design as supports, then we have the following table.

Table 2.2: Blocks of Fano Plane as supports and their related words.

Blocks as supports	**Words of the supports**
$\{1,2,5\}$	1100100
$\{1,3,6\}$	1010010
$\{1,4,7\}$	1001001
$\{2,3,7\}$	0110001
$\{2,4,6\}$	0101010
$\{3,4,5\}$	0011100
$\{5,6,7\}$	0000111

The words of the right side of the table above are the codewords of $[7,4,3]$-Hamming code of weight 3. All codewords can be seen in the Table 2.1.

Example 2.3.5. Codewords of extended $[8,4,4]$-Hamming code are composed of adding parity check bits at the end of each codeword of $[7,4,3]$-Hamming code. Then we have all codewords of $[8,4,4]$-Hamming code have weight 4 except the codewords $\bar{0}$ and $\bar{1}$. Now, we will give the supports of these codewords of weight 4 in Table 2.3 and we will call these supports as blocks of a design.

Table 2.3: Codewords of weight 4 of $[8,4,4]$-Hamming code and their supports.

Codewords of weight 4	**Supports**
11110000	$\{1,2,3,4\}$
11001001	$\{1,2,5,8\}$
10100101	$\{1,3,6,8\}$
00111001	$\{3,4,5,8\}$
01010101	$\{2,4,6,8\}$
10010011	$\{1,4,7,8\}$
01101100	$\{2,3,5,6\}$
01100011	$\{2,3,7,8\}$
10101010	$\{1,3,5,7\}$
11000110	$\{1,2,6,7\}$
10011100	$\{1,4,5,6\}$
01011010	$\{2,4,5,7\}$
00001111	$\{5,6,7,8\}$
00110110	$\{3,4,6,7\}$

All of the codewords in $[8,4,4]$-Hamming code have length 8, so let the point set $P = \{1,2,3,4,5,6,7,8\}$. It can be seen that every 3 point of P, that is $t = 3$, is exactly in 1 block, that is $\lambda = 1$. Therefore, each block has 4 point, that is $k = 4$. So, extended $[8,4,4]$-Hamming code is 3-$(8,4,1)$ design.

2.3.1. Assmus-Mattson Theorems

By [18], a code and its dual has specific forms of weight distributions, then the codewords of specified weights in both the code and its dual hold t-designs. In the view of this, if a code does not have the weight distribution in a specific form, we can not be sure whether codewords of a certain weight in a code hold a t-design. For insurance, we count the number of each t-sets contained in each word by a computer program. This procedure takes much time and work, and it may not be possible to apply for some large codes containing a design. Indeed, the Assmus-Mattson theorem is the main tool in determining designs in codes.

Firstly, we will state the Assmus-Mattson theorem for binary codes.

Theorem 2.3.1.1. (Assmus-Mattson) Let C be a binary $[n, k, d]$ code. Suppose $C^{\perp}$ has minimum weight $d^{\perp}$. Suppose that $A_i = A_i(C)$, and $A_i^{\perp} = A_i(C^{\perp})$ for $0 \leq i \leq n$, are the weight distributions of C and $C^{\perp}$, respectively. Fix a positive integer t with $t < d$, and let s be the number of i with $A_i^{\perp} \neq 0$ for $0 < i \leq n - t$. Suppose $s \leq d - t$. Then:

i. the vectors of weight i in C hold a t-design provided $A_i \neq 0$ and $d \leq i \leq n$, and
ii. the vectors of weigth i in $C^{\perp}$ hold a t-design provided $A_i^{\perp} \neq 0$ and $d^{\perp} \leq i \leq n - t$.

Then, the next theorem is extension of the Theorem 2.3.1.1 to an arbitrary field $\mathbb{F}_q$.

Theorem 2.3.1.2. (Assmus-Mattson) Let C be an $[n, k, d]$ code over $\mathbb{F}_q$. Suppose $C^{\perp}$ has minimum weight $d^{\perp}$. Let w be the largest integer with $w \leq n$ satisfying

$$w - \left\lfloor \frac{w + q - 2}{q - 1} \right\rfloor < d$$

(So $w = n \, when \, q = 2$.) Define $w^{\perp}$ analogously using $d^{\perp}$. Suppose that $A_i = A_i(C)$, and $A_i^{\perp} = A_i(C^{\perp})$ for $0 \leq i \leq n$, are the weight distributions of C and $C^{\perp}$, respectively. Fix a positive integer t with $t < d$, and let s be the number of i with $A_i^{\perp} \neq 0$ for $0 < i \leq n - t$. Suppose $s \leq d - t$. Then:

i. the vectors of weight i in C hold a t-design provided $A_i \neq 0$ and $d \leq i \leq w$, and
ii. the vectors of weigth i in $C^{\perp}$ hold a t-design provided $A_i^{\perp} \neq 0$ and $d^{\perp} \leq i \leq \min\{w^{\perp}, n - t\}$.

Example 2.3.1.3. Let the code $C = \{0000000, 1111000, 1100100, 1010010, 0110001, 0011100, 0101010, 1001001, 0110110, 1010101, 1100011, 1001110, 0101101, 0000111, 0011011, 1111111\}$ and therefore $C^{\perp} = \{0000000, 1001110, 0101101, 0011011, 1100011, 1010101, 0110110, 1111000\}$.

First, we check that C satisfies the conditions in the theorem. Here, C has minimum weight 3 and $C^{\perp}$ has minimum weight 4. Let $t = 2$, then $s = 1$ and $i = 4$. Since bounds in the first condition of the theorem is satisfied then the codewords of weight 4 in C holds a 2-design. Let

$P = \{1,2,3,4,5,6,7\}$ and observe that codewords of weight 4 in the dual of C are the codewords of weigth 4 in C. Then, we can constuct their supports as a block set B,

$$B = \{\{1,4,5,6\},\{2,4,5,7\},\{3,4,6,7\},\{1,2,6,7\},\{1,3,5,7\},\{2,3,5,6\},\{1,2,3,4\}\}.$$

So, we can see that (P, B) is a 2-$(7,4,2)$ design.

By the Assmus-Mattson Theorems, it is easy to find designs from the codes which are extremal binary doubly-even self-dual codes, we gave the definitions and properties of these codes in the Section 2.1, since extremal self-dual codes provide maximum d; and doubly-even codes provide minimum number of non-zero weights. Moreover, self-duality implies that we know the weight distributions of both the code and its dual. More information about constructions of designs can be found in [19].

By [14], the extremal doubly-even codes hold t-designs as a consequence of the theorem of Assmus-Mattson.

3. RESULTS

In this chapter, we present the results of our Mathematica program in Section 3.1. This program returns the lists of weigths of codes which give 1-design, if exists. We denote these weights by w. We run this program for each $2 \leq k \leq 14$ and for each $k < n \leq 15$, where k is the dimension and n is the length of a code.

In Section 3.2, we present our results about the existence of linear 1-design embedded codes by putting necessary conditions on the general values of n and k. We will see that the values presented in these tables are special cases of our results.

3.1. RESULTS OBTAINED FROM THE PROGRAM

In this section, we present the numerical results of our program in Tables 3.1- 3.13 for each $2 \leq k \leq 14$ and for each $k < n \leq 15$.

Table 3.1: The values of w for $k = 2$ and $3 \leq n \leq 15$.

For $k = 2$	
Values of n	**Values of w**
3	2
4	2
5	-
6	3, 4
7	-
8	4
9	6
10	5
11	-
12	6, 8
13	-
14	7
15	10

Table 3.2: The values of w for $k = 3$ and $4 \leq n \leq 15$.

For $k = 3$	
Values of n	**Values of w**
4	2
5	-
6	2, 3, 4
7	4
8	4
9	3, 6
10	5
11	-
12	6,8
13	-
14	7,8
15	10

Table 3.3: The values of w for $k = 4$ and $5 \leq n \leq 15$.

For $k = 4$	
Values of n	**Values of w**
5	2, 4
6	2, 3, 4
7	3, 4
8	2, 4, 6
9	3, 4, 6
10	4, 5, 6
11	-
12	6, 8
13	-
14	7, 8
15	6, 10

Table 3.4: The values of w for $k=5$ and $6 \leq n \leq 15$.

For $k = 5$	
Values of n	**Values of w**
6	2, 3, 4
7	3, 4
8	2, 4, 6
9	3, 4, 5, 6
10	4, 5, 6, 8
11	-
12	4, 6, 8
13	-
14	7, 8
15	5, 6, 8, 10

Table 3.5: The values of w for $k = 6$ and $7 \leq n \leq 15$.

For $k = 6$	
Values of n	**Values of w**
7	2, 4, 6
8	2, 4, 6
9	2, 4, 6
10	2, 3, 4, 5, 6, 7, 8
11	-
12	2, 3, 4, 5, 6, 7, 8, 9
13	8
14	7, 8
15	5, 6, 8, 10

Table 3.6: The values of w for $k = 7$ and $8 \leq n \leq 15$.

For $k = 7$	
Values of n	**Values of w**
8	2, 4, 6
9	2, 3, 4, 5, 6, 7
10	3, 5, 7
11	6
12	3, 4, 5, 6, 7, 8, 9
13	7
14	4, 6, 7
15	6, 10, 12

Table 3.7: The values of w for $k = 8$ and $9 \leq n \leq 15$.

For $k = 8$	
Values of n	**Values of w**
9	2, 4, 6, 8
10	2, 4, 5, 6, 8
11	-
12	5
13	5
14	4, 6, 7, 8, 10
15	4, 6, 7, 8, 10, 14

Table 3.8: The values of w for $k = 9$ and $10 \leq n \leq 15$.

For $k = 9$	
Values of n	**Values of w**
10	2, 4, 5, 6, 8
11	3,8
12	6
13	-
14	7
15	4, 5, 6, 8, 9, 10, 12

Table 3.9: The values of w for $k = 10$ and $11 \leq n \leq 15$.

For $k = 10$	
Values of n	**Values of w**
11	2, 3, 4, 6, 7, 8, 10
12	2, 4, 6, 8, 10
13	6, 7, 9
14	6, 7
15	4, 6, 8, 10, 12

Table 3.10: The values of w for $k = 11$ and $12 \leq n \leq 15$.

For $k = 11$	
Values of n	**Values of w**
12	2, 4, 6, 8, 10
13	5, 7, 8
14	2, 4, 6, 7, 8, 10, 12
15	5, 9, 10

Table 3.11: The values of w for $k = 12$ and $13 \leq n \leq 15$.

For $k = 12$	
Values of n	**Values of w**
13	2, 4, 6, 8, 10 12
14	7
15	8

Table 3.12: The values of w for $k = 13$ and $14 \leq n \leq 15$.

For $k = 13$	
Values of n	**Values of w**
14	2, 4, 6, 7, 8, 10, 12
15	2, 4, 6, 8, 10, 12

Table 3.13: The values of w for $k = 14$ and $n = 15$.

When $k = 14$	
Values of n	**Values of w**
15	2, 4, 6, 7, 8, 10, 12

When we examine the data presented in Tables 3.1- 3.4, we see that there is no code with paramaters $[6, k, d]$, $k \leq 5$, whose codewords of weight 5 form a 1-design. The reason behind this can be given as follows: Let us take a codeword of weight 5 from the code and let it be in the generating matrix. Without loss of generality, let us take this as $(1\ 1\ 1\ 1\ 1\ 0)$. Then to get 1-design, the number of appearance of every positions in codewords must be equal. Thus if this codeword is in the code, then $(1\ 1\ 1\ 1\ 0\ 1)$ should be in the code as well. Similarly, $(1\ 1\ 1\ 0\ 1\ 1)$, $(1\ 1\ 0\ 1\ 1\ 1)$, $(1\ 0\ 1\ 1\ 1\ 1)$, $(0\ 1\ 1\ 1\ 1\ 1)$ are also in the code. Since these are the all codewords in the code and they are linearly independent, the generating matrix for this code is

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

Consequently, such a code with codewords of weight 5 gives a 1-design. However, in this case $C = \mathbb{F}_2^6$ and so C has dimension 6 which contradicts the fact that $k < 6$. We generalize this observation in Lemma 3.2.5.

Moreover, there exists a binary code of length 6 whose codewords of weight 4 form a 1-design. The reason behind this is given as follows: Let us take a code with the following generating matrix

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

Then all codewords of this code are 000000, 111100, 110011, 001111. Codewords of weight 4 construct a 1-design since the number of appearance of each position in these codewords is 2. Then parameters of this design is 1-$(6, 4, 2)$. We generalize this observation in Lemma 3.2.7.

According to Table 3.1, that is, when $k = 2$ and $n \leq 15$, if n is odd and not divisible by 3, then the codewords of weight s can not form a 1-design for any s with $1 \leq s < n$. A generalization of this observation is presented in Lemma 3.2.8.

3.2. DESIGN EMBEDDED CODES

In this section, we define a code from which a design can be obtained, a '*design embedded code*'. By this definition, observe that Assmus-Mattson Theorems produce design embedded codes. Moreover, by Theorem 2.3.3, perfect codes are also design embedded codes. In both cases there are many restrictive conditions on the values of n, t, and also on the code itself. These conditions make it difficult to find design embedded codes. So, it is interesting to find linear design embedded codes without using perfect codes and Assmus-Mattson theorems.

Observe that, if we take any design, and if we take the blocks of this design as supports of the binary vectors of a code then this code will be a nonlinear design embedded code since it will not have $\bar{0}$ codeword. So, by using this method, many nonlinear design embedded codes can be found.

Now, we give the lemmas about the existence of linear 1-design embedded codes without using perfect codes and Assmus-Mattson theorems. We will see that the special cases of Lemmas 3.2.5- 3.2.8 can be seen in Tables 3.1.1-3.1.13.

Lemma 3.2.1. Suppose C_1 and C_2 are two codes with parameters $[n_1, k_1, d]$ and $[n_2, k_2, d]$, respectively. Assume that the supports of codewords of minimum weight in C_1 and C_2 form a

1-design, with $\lambda_1 = \lambda_2$. Then $C_1 \oplus C_2$ has parameters $[n_1 + n_2, k_1 + k_2, d]$, and it is also a 1-design embedded code.

Proof. The direct sum of C_1 and C_2 is

$$C_1 \oplus C_2 = \{(\overline{c_1}, \overline{c_2}\ )|\ \overline{c_1} \in C_1, \overline{c_2} \in C_2\ \} \tag{3.1}$$

If G_1 is a generating matrix for C_1 and G_2 is a generating matrix for G_2, then a generating matrix for $C_1 \oplus C_2$ can be taken as

$$\begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \tag{3.2}$$

Let the codewords $\overline{c_1}, \overline{c_2}$ have weight d. Then, in $C_1 \oplus C_2$, the codewords $(\overline{c_1}, \bar{0})$ and $(\bar{0}, \overline{c_2})$ have also weight d, and the weight d just comes from these two types of vectors, and for these codewords $\lambda_1 = \lambda_2$, then $C_1 \oplus C_2$ is a 1-design embedded code. □

Lemma 3.2.2. Let C_1, C_2, … , C_m be linear codes with parameters $[n_1, k_1, d]$, $[n_2, k_2, d]$, … , $[n_m, k_m, d]$, respectively. Assume that the supports of codewords of minimum weight in C_i, $i \in \{1, \dots, m\}$ form a 1-design with $\lambda_1 = \lambda_2 = \cdots = \lambda_m$. Then $C_1 \oplus C_2 \oplus \dots \oplus C_m$ has parameter $[n_1 + n_2 + \cdots + n_m, k_1 + k_2, + \cdots + \ k_m, d]$, and it is also a 1-design embedded code.

Proof. Direct sum of C_i's for $i \in \{1, \dots, m\}$ is

$$C_1 \oplus C_2 \oplus \dots \oplus C_m = \{(\overline{c_1}, \overline{c_2}, \dots\ , \ \overline{c_m}) \colon \overline{c_1} \in C_1,\ \overline{c_2} \in C_2, \dots, \overline{c_m} \in C_m\} \tag{3.3}$$

If G_i is a generating matrix for C_i for $i \in \{1, \dots, m\}$ then,

$$\begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & G_m \end{bmatrix} \tag{3.4}$$

is a generating matrix for $C_1 \oplus C_2 \oplus \dots \oplus C_m$.

Let the codewords $\overline{c_1}, \overline{c_2}, \dots\ , \ \overline{c_m}$ have weight d, where $\overline{c_1} \in C_1,\ \overline{c_2} \in C_2, \dots, \overline{c_m} \in C_m$. Then in $C_1 \oplus C_2 \oplus \dots \oplus C_m$, the codewords $(\overline{c_1}, \bar{0}, \dots, \bar{0})$, $(\bar{0}, \overline{c_2}, \bar{0} \dots, \bar{0})$, $(\bar{0}, \dots \bar{0}, \overline{c_m})$ have also weight d, and the weight d just comes from such vectors, and for these codewords $\lambda_1 = \lambda_2 = \cdots = \lambda_m$, then $C_1 \oplus C_2 \oplus \ldots \oplus C_m$ is a 1-design embedded codes. □

Corollary 3.2.3. (General family of 1-design embedded codes) Consider C be a binary repetition code of length k, $k \geq 1$. Then, $C \oplus C \oplus \ldots \oplus C$, the direct sum of m copies of C, is a 1-design embedded linear code with the parameters $[mk, m, k]$. The weights of all the nonzero codewords belong to the set $\{k, 2k, \ldots, mk\}$, and the supports of the codewords of all weights form 1-designs.

Proof. Assume that C is a repetition code of parameter $[k, 1, k]$. Then,

$$C \oplus C \oplus \ldots \oplus C = \{(\overline{c_1}, \overline{c_2}, \ldots, \overline{c_m}) \mid \overline{c_1}, \overline{c_2}, \ldots, \overline{c_m} \in C\} \tag{3.5}$$

If C has a generating matrix G,

$$\begin{bmatrix} G & 0 & \cdots & 0 \\ 0 & G & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & G \end{bmatrix} \tag{3.6}$$

is generating matrix for $C \oplus C \oplus \ldots \oplus C$.

Let the length of the direct sum $C \oplus C \oplus \ldots \oplus C$ be n and let the set $\{1, 2, \ldots, n\}$ be a point set, and the supports of the codewords have weight d be a block set. Let λ_d denotes the number of appearance of every element in these blocks.

Codewords of weight k in the direct sum $C \oplus C \oplus \ldots \oplus C$ are produced in this way; codewords of weight k just come from one code, the other $m-1$ codes give their $\bar{0}$ codewords. So we have $\binom{m}{1}$ possible codewords of weight k. Moreover, each point appears in the blocks 1 times, that is $\lambda_k = 1$.

Codewords of weight $2k$ in the direct sum come from each pair of codewords of weight k in the code C. We have $\binom{m}{2}$ possible codewords of weight $2k$. Moreover, for finding λ_{2k} we hold one of these codewords, then we have $\binom{m-1}{1}$ different codewords of weight k, and this number gives us the value of λ_{2k}.

Now, we prove that the values of λ_d's where $d = \{k, 2k, \dots, mk\}$ are the same as the coefficients of $(x+1)^{m-1}$. Observe that, it does not depend on the value of k. On the other hand d does not depend on the value of m, it is just related with the value of k. Now, for fixed value of k, we give the proof by induction on m.

When $m = 1$, we have $C = [k, 1, k]$ with minimum distance k and length k. The generating matrix for C is a $1 \times k$ matrix $[11 \dots 1]$. Thus the code C has two codewords, $\bar{0}$ and $\bar{1}$. Here, the point set is $\{1, 2, \dots, k\}$ and the block set is the support of the nonzero codeword, that is $\{1, 2, \dots, k\}$. So, each point appears 1 times in the block, that is $\lambda_1 = 1$. Hence, the assertion is true for $m = 1$.

Let $n \in Z^+$ be given and assume that the assertion is true for $m = n$. Then, we will examine it for $m = n + 1$. Let G denotes the generating matrix for the direct sum of the m copies of C when $m = n$, then the generating matrix for the direct sum of the m copies of C when $m = n + 1$ is $\begin{bmatrix} G & 0 \dots 0 \\ 0 \dots 0 & 1 \dots 1 \end{bmatrix}$. For the direct sum of n copies of the code C, let λ_d denotes the number of appearance of every point in the block set where each block has d points. Here, the point set is $\{1, 2, \dots, mk\}$. When $m = n + 1$, we denote β_d's for the number of appearance of every point in the block set where each block has d points. Then, for the value β_k there is no effect of adding a code to direct sum of n codes, so β_k is equal to λ_k, that is $\binom{n-1}{0} = \binom{n}{0}$. Now, let $w(d)$ denotes the codewords of weight d in the direct sum of $m - 1$ copies of C. The value β_{2k} comes from $w(2k)$ and the codewords where $\bar{1}$ is added at the end of the each codewords in $w(k)$, $\binom{n-1}{1} + \binom{n-1}{0}$. This sum is equal to $\binom{n}{1}$ from Pascal identity. The value β_{3k} comes from $w(3k)$ and the codewords where $\bar{1}$ is added at the end of the each codewords in $w(2k)$, $\binom{n-1}{2} + \binom{n-1}{1} = \binom{n}{2}$. If we continue this procedure, we get $\beta_{nk} = \binom{n-1}{n-1} + \binom{n-1}{n-2} = \binom{n}{n-1}$ from $w(nk)$ and codewords where $\bar{1}$ is added at the end of the each codewords in $w((n-1)k)$. Lastly $\beta_{(n+1)k} = \binom{n-1}{n-1} = \binom{n}{n}$ codewords which is added $\bar{1}$ on the end of the each codewords in $w(nk)$. Thus, values of β_d's are the coefficients of $(x+1)^{m-1}$ when $m = n + 1$. So, the assertion is proven, and thus the Corollary 3.2.3 is proven. □

Now, we will give some values of λ_d for some values of k and m.

For $k = 1$ and $m = 1$, $C = \{0, 1\}$. So, $d = 1$ and $\lambda_1 = 1$.

For $k = 1$ and $m = 2$, we have $C = \{0, 1\}$ and $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is the generating matrix for the code $C \oplus C$. So, $C \oplus C = \{00, 10, 01, 11\}$. Then, for $d = 1$, $\lambda_1 = 1$, and for $d = 2$, $\lambda_2 = 1$.

For $k = 3$ and $m = 3$, $C = \{000, 111\}$ and $\begin{bmatrix} 111000000 \\ 000111000 \\ 000000111 \end{bmatrix}$ is the generating matrix for the code $C \oplus C \oplus C$. So, $C \oplus C \oplus C = \{000000000, 111000000, 000111000, 000000111, 111111000, 111000111, 000111111, 111111111\}$. Thus, for $d = 3$, we have $\lambda_3 = 1$; and for $d = 6$, we have $\lambda_6 = 2$; for $d = 9$, we have $\lambda_9 = 1$.

Now, we will give a table of λ_d's for $k \in \{1, 2, 3, 4, 5\}$ and $m \in \{1, 2, 3, 4, 5\}$.

Table 3.14: Values of λ_d's for different values of k and m.

m \ k	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$m = 1$	$\lambda_1 = 1$	$\lambda_2 = 1$	$\lambda_3 = 1$	$\lambda_4 = 1$	$\lambda_5 = 1$
$m = 2$	$\lambda_1 = 1$, $\lambda_2 = 1$	$\lambda_2 = 1$, $\lambda_4 = 1$	$\lambda_3 = 1$, $\lambda_6 = 1$	$\lambda_4 = 1$, $\lambda_8 = 1$	$\lambda_5 = 1$, $\lambda_{10} = 1$
$m = 3$	$\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 1$	$\lambda_2 = 1$, $\lambda_4 = 2$, $\lambda_6 = 1$	$\lambda_3 = 1$, $\lambda_6 = 2$, $\lambda_9 = 1$	$\lambda_4 = 1$, $\lambda_8 = 2$, $\lambda_{12} = 1$	$\lambda_5 = 1$, $\lambda_{10} = 2$, $\lambda_{15} = 1$
$m = 4$	$\lambda_1 = 1$, $\lambda_2 = 3$, $\lambda_3 = 3$, $\lambda_4 = 1$	$\lambda_2 = 1$, $\lambda_4 = 3$, $\lambda_6 = 3$, $\lambda_8 = 1$	$\lambda_3 = 1$, $\lambda_6 = 3$, $\lambda_9 = 3$, $\lambda_{12} = 1$	$\lambda_4 = 1$, $\lambda_8 = 3$, $\lambda_{12} = 3$, $\lambda_{16} = 1$	$\lambda_5 = 1$, $\lambda_{10} = 3$, $\lambda_{15} = 3$, $\lambda_{20} = 1$
$m = 5$	$\lambda_1 = 1$, $\lambda_2 = 4$, $\lambda_3 = 6$, $\lambda_4 = 4$, $\lambda_5 = 1$	$\lambda_2 = 1$, $\lambda_4 = 4$, $\lambda_6 = 6$, $\lambda_8 = 4$, $\lambda_{10} = 1$	$\lambda_3 = 1$, $\lambda_6 = 4$, $\lambda_9 = 6$, $\lambda_{12} = 4$, $\lambda_{15} = 1$	$\lambda_4 = 1$, $\lambda_8 = 4$, $\lambda_{12} = 6$, $\lambda_{16} = 4$, $\lambda_{20} = 1$	$\lambda_5 = 1$, $\lambda_{10} = 4$, $\lambda_{15} = 6$, $\lambda_{20} = 4$, $\lambda_{25} = 1$

Lemma 3.2.4. Let C be a binary code of the parameters $[n, n-1,2]$, consisting of all even weight vectors of length n. It is also described as $D^{\perp}$, where $D = \langle 11 \ldots 1 \rangle$ is the repetition code. Then C is a 1-design embedded code.

Proof. The set $D_n = \{(00 \ldots 0), (11 \ldots 1)\}$ is the binary repetition code of length n.

${D_n}^{\perp} = \{v \in \mathbb{F}_2^n : \langle v, (00 \ldots 0) \rangle = 0, \langle v, (11 \ldots 1) \rangle = 0\}$. The first condition, $\langle v, (00 \ldots 0) \rangle$, holds for all words $v \in \mathbb{F}_2^n$. The second condition, $\langle v, (11 \ldots 1) \rangle = 0$, means that $1 + 1 + \cdots + 1 = 0$ in $\mathbb{F}_2$, i.e. v is from binary even weight code of length n. Thus $D^{\perp} = C$.

Now, we can prove this lemma in two different ways.

1. By using Assmus-Mattson Theorem for binary case:

By the theorem of Assmus-Mattson we have $[n, n-1,2]$, $d = 2$. Then we have the following inequalities $t < 2$ and $s \leq 2 - t$, and C has codewords of weight $0, 2, 4, \ldots, 2\left\lfloor \frac{n}{2} \right\rfloor$, only.

The maximum value of t which satisfies these conditions is $t = 1$, in which case $s = 1$. Weight distribution of $C^{\perp}$ is 0 and n, only.

i. According to the first case of the theorem, vectors of weight i in C hold a t-design provided $A_i \neq 0$ and $2 \leq i \leq n$. Then, codewords of weights $2, 4, \ldots, 2\left\lfloor \frac{n}{2} \right\rfloor$ in C hold a 1-design.

ii. Second case of the theorem can not be applied since the bounds are not appropriate.

2. Without using Assmus-Mattson Theorem:

In this way, we have a good consequence in comparison to the first way since now we will not give only 1-design embedded codes, but we will indicate that these are also 2-designs, 4-designs, etc.

All supports of codewords of length n is from the set $\{1, 2, \ldots, n\}$. We will take the even-element subsets of this set. For instance, we will take the 2-element subsets of this set , $\binom{n}{2}$, then we know that the number of appearance of any element in this subsets, $t = 1$, that is $\binom{n-1}{1}$. This number is equal the λ_1, and with these values we get the parameters of the design, $1\text{-}\left(n, 2, \binom{n-1}{1}\right)$.

If $t = 2$, $\lambda_2 = 1$, since blocks are 2-element subsets, then the parameter of the design is $2\text{-}(n, 2,1)$.

Let us take the 4-element subsets of the set $\{1,2, \dots, n\}$. For $t = 4$, $\lambda_4 = 1$, then the design is of the parameter $4\text{-}(n, 4, 1)$. Therefore, this design is also 3-design, 2-design and 1-design, because we can find the number of subsets, i.e. blocks, that have the same 3 elements, 2 elements and 1 element. Their parameters are $3\text{-}\left(n, 4, \binom{n-3}{1}\right)$, $2\text{-}\left(n, 4, \binom{n-2}{2}\right)$, $1\text{-}\left(n, 4, \binom{n-1}{3}\right)$, respectively. □

Lemma 3.2.5. Assume that C is any binary code of parameters $[2n, k, d]$, where $k < 2n$. Then the codewords of weight $2n - 1$ cannot form a 1-design.

Proof. Let us take a codeword of weight $2n - 1$ and let it be in generating matrix, let us take it as $(1\ 1\ 1 \dots 1\ 1\ 0)$, without loss of generality. Then to get 1-design, the number of appearance of every positions in codewords must be equal. Thus if this is in the code, then $(1\ 1\ 1 \dots 1\ 0\ 1)$ should be in the code, and also $(1\ 1\ 1 \dots 0\ 1\ 1), \dots, (0\ 1\ 1 \dots 1\ 1\ 1)$ should be in the code. Since this code is linear, it has a generating matrix. Since these are all in the code and they are linearly independent then the generating matrix can be taken as

$$G = \begin{bmatrix} 1\ 1\ 1 \dots 1\ 1\ 0 \\ 1\ 1\ 1 \dots 1\ 0\ 1 \\ \vdots \\ 1\ 0\ 1 \dots 1\ 1\ 1 \\ 0\ 1\ 1 \dots 1\ 1\ 1 \end{bmatrix}.$$

Then the number of appearance of every position is $2n - 1$, but the generating matrix G is of dimension $2n$. It contradicts to the assumption that the dimension $k < 2n$. □

Lemma 3.2.6. Let n be even. Then, there exists a binary code of length n whose codewords of weight $n/2$ form a 1-design.

Proof. Let us take a code C of parameter $[2k, 2, k]$. Suppose that the generating matrix of C consists of the following codewords: $\overline{c_1} = 1 \dots 10 \dots 0$, it has k copies of 1 and k copies of 0, and $\bar{c_2} = 0 \dots 01 \dots 1$, it has k copies of 0 and k copies of 1. Now it is clear that these two codewords of weight k always construct a 1-design. □

Lemma 3.2.7. Let n be divisible by 3. There exists a binary code of length n, whose codewords of weight $2n/3$ form a 1-design.

Proof. Assume that C is a code with its generating matrix consists of following two codewords:

$(1 \dots 11 \dots 10 \dots 0\ )$: 1's are in $2k$ positions and 0's are in k positions consequtively, and

$(1 \dots 10 \dots 01 \dots 1\ )$: 1's are in k positions, 0's are in k positions, and 1's are in k positions, conseqently.

All codewords of this code are $(00 \dots 0)$, $(1 \dots 11 \dots 10 \dots 0)$, $(1 \dots 10 \dots 01 \dots 1)$, $(0 \dots 01 \dots 11 \dots 1)$.

Weights of the codewords are 0 and $2k$ only. The number of appearance of every position is 2, i.e $\lambda = 2$. It constructs a 1-design of parameters 1-$(n, 2k, 2)$. □

Lemma 3.2.8. Consider C be a binary code of parameters $[n, 2, d]$, where n is odd and not divisible by 3. Then, for any s with $d \leq s < n$, the codewords of weight s cannot form a 1-design.

Proof. The dimension of the code C is 2. It means there exist two codewords in the generating matrix of C. Let us denote these two codewords as $\bar{c_1}$ and $\bar{c_2}$. All codewords of C are $\bar{0}$, $\bar{c_1}$, $\bar{c_2}$, and $\bar{c_1} + \bar{c_2}$. We want to get 1-design from this code, for this there are two cases of the weights of the codewords.

1. In the first case, $\bar{c_1}$, $\bar{c_2}$, $\bar{c_1} + \bar{c_2}$ all have the same weight or $\bar{c_1}$ and $\bar{c_2}$ must be complements of each other. In the first case,
    i. For $\lambda = 1$, we have $w(\bar{c_1}) = w(\bar{c_2}) = w(\bar{c_1} + \bar{c_2}) < n$, $\bar{c_1} = \big(1(a)\ 0(b)\ 0(c)\big)$, $\bar{c_2} = \big(\ 0(a)\ 1(b)\ 0(c)\big)$, $\bar{c_1} + \bar{c_2} = \big(0(a)\ 0(b)\ 1(c)\big)$, where $1(a)$ denotes that there are a copies of 1 and $0(a)$ denotes that there are a copies 0, and similarly for $0(b)$, $0(c)$, $1(b)$, $1(c)$. However, there does not exist any $\bar{c_1} + \bar{c_2}$ in this form.

ii. For $\lambda = 2$, we have $\overline{c_1} = (1(a)\ 1(b)\ 0(c))$, $\overline{c_2} = (1(a)\ 0(b)\ 1(c))$, $\overline{c_1} + \overline{c_2} = (0(a)\ 1(b)\ 1(c))$. Since we assume that the weights of $\overline{c_1}$, $\overline{c_2}, \overline{c_1} + \overline{c_2}$ are equal, we have $a + b = b + c = a + c$. Therefore $a = b = c$ and $n = 3a$. This is a contradiction to the condition of $n \neq 3k$ in Lemma 3.1.8.

2. For the second case, let $w(\overline{c_1}) = w(\overline{c_2}) < n$, and $w(\overline{c_1} + \overline{c_2}) \neq w(\overline{c_1})$. Since the weights of $\overline{c_1}$, $\overline{c_2}$ are equal, we have $\lambda = 1$. Then $\overline{c_1} = (1(a)\ 0(b))$, $\overline{c_2} = (0(a)\ 1(b))$ which gives $\overline{c_1} + \overline{c_2} = (11 \ldots 1)$. Therefore $w(\overline{c_1} + \overline{c_2}) = w(\overline{c_1}) + w(\overline{c_2})$ and $n = w(\overline{c_1}) + w(\overline{c_2}) = 2w(\overline{c_1})$. Thus n is even which is a contradiction to the fact that n is odd. □

4. DISCUSSION

In this thesis, we study on codes and combinatorial designs, and also the connections between codes and combinatorial designs.

There are two natural problems in this area: constructing codes from designs and constructing designs from codes. The incidence matrix of a design can be taken as generating matrix or parity check matrix for a code. By this way, codes can be obtained from designs without difficulty. Conversely, a design can be obtained from codes but it is more complex problem than obtaining codes from designs. Because, designs require special conditions such as all t-subsets must be in precisely λ blocks and the number of blocks is bounded by the values of t, v, k.

We define a code as a design embedded code if this code contains a design. By Theorem 2.3.3, we know that perfect codes are design embedded codes. However, by the definition of a perfect code, we can only obtain a design from a perfect code when the minimum distance is odd and the Hamming bound of the code is attained. Assmus-Mattson Theorems is an another way to get designs from codes but these theorems have some restrictive conditions. According to these theorems, a code and its dual should have specific forms of weight distributions and also they have a restrictive condition on t. So, it is interesting to find design embedded codes without using Assmus-Mattson Theorems and perfect codes. In this manner, we study on especially 1-design embedded codes.

We present the numerical results obtained from our computer program implemented in Mathematica. Then, we give results related to 1-design embedded codes. Special cases of these results can be seen in the numerical results of the program.

We see in the proof of Lemma 3.2.4 that the second case of the Assmus-Mattson theorem for binary case do not work because the bounds are not appropriate. In this proof, the first case of the Assmus-Mattson theorem produces 1-designs. However, in this poof, we also give data about 2-design embedded codes, 4-design embedded codes, etc. We also discuss some parameters of these designs in codes.

In Lemma 3.2.6 and Lemma 3.2.7, we give the knowledge about 1-design embedded codes. Moreover, there is no information about the weight distributions of the codes and also we do not know these codes are perfect or not.

5. CONCLUSION AND RECOMMENDATIONS

In this thesis, we obtain some results about 1-design embedded codes by studying on codes and combinatorial designs and the relations between codes and combinatorial designs.

We first overview the basics of Coding Theory and Combinatorial Design Theory. We present basic definitions, theorems and related examples in these theories. Then, we analyse the connections between codes and combinatorial designs and present the Assmus-Mattson Theorems. We give the numerical results of our Mathematica program, which is given in Appendix. Then, we define 'design embedded codes' and present our results about 1-design embedded codes without using Assmus-Mattson Theorems and perfect codes in Lemmas 3.2.1-3.2.8.

As a continuation of this study, one can develop some further methods to construct 1-design embedded codes without using Assmus-Mattson Theorems and perfect codes. It will be more intresting and challenging to obtain t-design embedded codes when $t \geq 2$ without using Assmus-Mattson Theorems and perfect codes.

REFERENCES

[1]. Shannon, C. E., 1948, A Mathematical Theory of Communication, *Bell Syst. Tech. J.*, 27(3), 379-423.

[2]. Hamming, R. W., 1950, Error Detected and Error Correcting Codes, *The Bell System Technical Journal*, 29(2), 147-160.

[3]. Küçükçiftçi, S. and Özkan, S., 2004, Tasarımlar, *Matematik Dünyası*, 4, 28-34.

[4]. Kirkman, T. P., 1850, Query VI, *Lady's and Gentleman's Diary*, 147, 48.

[5]. Wilson, R., 2003, The Early History of Block Designs, *Rendiconti del Seminario Matematico di Messina*, 11(9), 267-276.

[6]. Colbourn, C. J. and Oorschot, P.C.V., 1989, Applications of Combinatorial Designs in Computer Science, *ACM Computing Surveys* 21(2), 223-250.

[7]. Colbourn, C. J., Dinitz, J. H. and Stinson, D. R., 1999, Applications of Combinatorial Designs to Communications, Cryptography, and Networking, *Surveys in combinatorics, 1999*, London Math. Soc. Lecture Note Series, 267, In: Lamb, J.D. and Preece, D. A. (ed.), Cambridge University Press, Cambridge, 37-100.

[8]. Naes, T., Brockhoff, P. B. And Tomic, O., 2010, Design of Experiments for Sensory and Consumer Data, *Statistics for Sensory and Consumer Science*, Part II, Wiley, United Kingdom, 181-191.

[9]. Du, D. and Hwang, F., 2006, Pooling Designs and Nonadaptive Group Testing: Important Tools for DNA Sequencing, *World Scientific*, Singapore.

[10]. Colbourn, C. J. and J. H. Dinitz, 1996, *CRC Handbook of Combinatorial Designs*, CRC Publishing Co., Boca Raton, Florida.

[11]. Alltop, W. O., 1966, On the Construction of Block Designs, *Journal of Combinatorial Theory*, 1(4), 501-502.

[12]. Huffman, W. C. and Pless, V., 2003, *Fundamentals of Error Correcting Codes*, Cambridge University Press, New York.

[13]. van Lint, J. H. and Wilson, R. M., 2001, *A Course in Combinatorics*, Cambridge University Press, New York.

[14]. Bouyuklieva, S., Varbanov, Z., 2011, Some Connections Between Self-Dual Codes, Combinatorial Designs and Secret Sharing Schemes, *Advances in Mathematics of Communications*, 5(2), 191-198.

[15]. Huffman, W. C. and Pless, V., 2003, *Designs*, *Fundamentals of Error Correcting Codes*, Chapter 8, Cambridge University Press, New York, 291-337.

[16]. van Lint, J. H. and Wilson, R. M., 2001, *Designs*, *A Course in Combinatorics*, Chapter 19, Cambridge University Press, New York, 215-243.

[17]. Assmus, E. F., J.R., Mattson, H. F., J.R., 1967, On Tactical Configurations and Error Correcting Codes, *Journal of Combinatorial Theory*, 2(3), 243-257.

[18]. Assmus, E. F., Mattson, H. F., 1969, New 5-Desings, *Journal of Combinatorial Theory*, 6(2), 122-151.

[19]. Stinson, D. R., 2004, *Combinatorial Designs: Constructions and Analysis*, Springer, New York.

APPENDICES

Appendix 1. Mathematica Programme for $n = 3$ and $k = 2$.

//In Mathematica, all following commands are written. Now, we give the commands that we //used. Explanations are begun with //.

//This command finds the support of a vector vec_.

```
Supp[vec_]:=Module[
   {i=1, temp={}},
   While[i≤ Length[vec], temp = If[vec[[i]]== 1, Join[temp,{i}], temp];
     i=i+1;
     ];
   Return[temp];
   ]
```

//The following command finds Hamming weight of a vector ve_.

```
Ham[ve_]:=Sum[ve[[i]],{i,1,Length[ve]}]
```

//The following command lists supports of the codes of Hamming weight d.

```
SuppList[code_,d_]:=Module[
   {i=1, temp={}},
   While[i≤ Length[code], temp = If[ Ham[code[[i]]] == d, Join[temp, {Supp[code[[i]]]}], temp];
     i=i+1;
     ];
```

Appendix 1(cont.)

```
   Return[temp];
   ]
```

//The following command gives the number of the blocks that contains the point x_.

```
Appearance[bl_, x_]:= Module[
   {i=1, temp=0},
   While[i≤ Length[bl], temp = If[ MemberQ[bl[[i]],x]== True,temp+1, temp];
     i=i+1;
     ];
   Return[temp];
   ]
```

//The following command checks whether all elements of a list are the same or not.

```
IsListEqual[li_]:= If[Product[li[[j]],{j,1,Length[li]}] ≠ 0 && Sum[(li-(GCD @@ li))[[i]], {i,1,Length[li]}] ==0, True, False]
```

//The following command cheks whether the numbers of all points of a code is equal or not.

```
IsOneDesign[Li_,n_]:=If[IsListEqual[Table[Appearance[Li,j],{j,1,n}]], True, False]
```

//The following command checks whether codewords weight s_ of a code form 1-design or not.

```
CodeDesign[code_, s_ ]:=If[ IsOneDesign[SuppList[code,s],Length[code[[1]]]], True, False]
```

//The following command gives a generating matrix randomly for given values of n, here //we take $n = 3$.

```
GenM:={{1,0, Random[Integer,{0,1}]},{0,1,Random[Integer,{0,1}]}}
```

Appendix 1(cont.)

GenM

//The following set is the output of the previous command. When you press the commend above you get a set randomly.

//{{1,0,1},{0,1,1}}

//The following command gives the matrix of dimension k which is given by user, here //$k = 2$.

V=Flatten[Table[{a,b}, {a, 0, 1},{b, 0, 1}], 1];

V

//The following list is the output of the previous command.

//{{0,0},{0,1},{1,0},{1,1}}

//In the following S is introduced.

S:= (B=GenM;A = Mod[V.B, 2]; {B, CodeDesign[A,2]})

//The following command gives the values of GenM and says codewords have weight 2 forms //whether a 1-design or not.

S

//The following is an output of the previous command.

{{{1,0,1},{0,1,1}},True}

//We give a number that we want in the following command, here it is 10, since we want to take //most of different GenM values.

Table[S, {i, 1,10}]

//The following is the output of the previous command.

Appendix 1(cont.)

{{{{1,0,0},{0,1,0}},False},{{{1,0,1},{0,1,1}},True},{{{1,0,0},{0,1,1}},False},{{{1,0,1},{0,1,0}},False}}

//It lists the elements of the list above which consists True.

```
WithTrue[Li_]:=Module[
  {i=1, temp={}},
  While[i≤ Length[Li], temp = If[ MemberQ[Li[[i]],True]== True,Append[temp, Li[[i]]],temp];
    i=i+1;
    ];
  Return[temp];
  ]
```

//Then we introduce the taken lists from the command Table[S, {i, 1,10}] as follows //Orn(values of n)(values of k).

Orn32:={{{{1,0,0},{0,1,0}},False},{{{1,0,1},{0,1,1}},True},{{{1,0,0},{0,1,1}},False},{{{1,0,1},{0,1,0}},False}}

//The following command lists the element which has True from Orn(values of n)(values of k).

WithTrue[Orn32]

{{{1,0,1},{0,1,1}},True}

//The following command gives the length of the previous output.

Length[%]

//The following is output of the previous command

Appendix 1(cont.)

1

//The following command lists the position of the True expression.

Table[Position[%19[[i]],True],{i,1,1}]
//Here the number 19 is comes from the commands number. When you press shift+enter, it //appears a number at the beginning of the command.

//The following list is output of the previous command and it shows codewords of weight 2 //forms a 1-design.

{{{2}}}

Appendix 2. Mathematica Programme for $n = 4$ and $k = 3$.

```
Supp[vec_]:=Module[

   {i=1, temp={}},

   While[i≤ Length[vec], temp = If[vec[[i]]== 1, Join[temp,{i}], temp];

     i=i+1;

     ];

   Return[temp];

   ]

Ham[ve_]:=Sum[ve[[i]],{i,1,Length[ve]}]

SuppList[code_,d_]:=Module[

   {i=1, temp={}},

   While[i≤ Length[code], temp = If[ Ham[code[[i]]] == d, Join[temp, {Supp[code[[i]]]}], temp];
```

Appendix 2(cont.)

```
    i=i+1;
    ];
  Return[temp];
  ]
Appearance[bl_, x_]:= Module[
  {i=1, temp=0},
  While[i≤ Length[bl], temp = If[ MemberQ[bl[[i]],x]== True,temp+1, temp];
    i=i+1;
    ];
  Return[temp];
  ]
IsListEqual[li_]:= If[Product[li[[j]],{j,1,Length[li]}] ≠ 0 && Sum[(li-(GCD @@ li))[[i]], {i,1,Length[li]}] ==0, True, False]
IsOneDesign[Li_,n_]:=If[IsListEqual[Table[Appearance[Li,j],{j,1,n}]], True, False]
CodeDesign[code_, s_ ]:=If[ IsOneDesign[SuppList[code,s],Length[code[[1]]]], True, False]
Flatten[Table[{a,b,c},{a,0,1},{b,0,1},{c,0,1}],2]
{{0,0,0},{0,0,1},{0,1,0},{0,1,1},{1,0,0},{1,0,1},{1,1,0},{1,1,1}}
GenM:={{1,0,0,Random[Integer,{0,1}]},{0,1,0,Random[Integer,{0,1}]},{0,0,1,Random[Integer,{0,1}]}}
GenM
```

Appendix 2(cont.)

```
{{1,0,0,1},{0,1,0,0},{0,0,1,1}}

V=Flatten[Table[{a,b,c}, {a, 0, 1},{b, 0, 1},{c, 0, 1}], 2];

V

{{0,0,0},{0,0,1},{0,1,0},{0,1,1},{1,0,0},{1,0,1},{1,1,0},{1,1,1}}

S:= (B=GenM;A = Mod[V.B, 2]; {B, CodeDesign[A,2], CodeDesign[A,3]})

S

{{{1,0,0,1},{0,1,0,0},{0,0,1,0}},True,False}

Table[S, {i, 1,10}]

{{{{1,0,0,1},{0,1,0,1},{0,0,1,1}},True,False},{{{1,0,0,0},{0,1,0,0},{0,0,1,1}},True,False},{{{1,0,0,0},{0,1,0,0},{0,0,1,0}},False,False},{{{1,0,0,1},{0,1,0,1},{0,0,1,0}},False,False},{{{1,0,0,1},{0,1,0,0},{0,0,1,0}},True,False},{{{1,0,0,0},{0,1,0,1},{0,0,1,1}},False,False}}

Orn43:={{{{1,0,0,1},{0,1,0,1},{0,0,1,1}},True,False},{{{1,0,0,0},{0,1,0,0},{0,0,1,1}},True,False},{{{1,0,0,0},{0,1,0,0},{0,0,1,0}},False,False},{{{1,0,0,1},{0,1,0,1},{0,0,1,0}},False,False},{{{1,0,0,1},{0,1,0,0},{0,0,1,0}},True,False},{{{1,0,0,0},{0,1,0,1},{0,0,1,1}},False,False}}

WithTrue[Li_]:=Module[

  {i=1, temp={}},

  While[i≤ Length[Li], temp = If[ MemberQ[Li[[i]],True]== True,Append[temp, Li[[i]]],temp];

    i=i+1;

    ];

  Return[temp];
```

Appendix 2(cont.)

```
  ]
WithTrue[Orn43]
{{{{1,0,0,1},{0,1,0,1},{0,0,1,1}},True,False},{{{1,0,0,0},{0,1,0,0},{0,0,1,1}},True,False},{{{1,0,0,1},{0,1,0,0},{0,0,1,0}},True,False}}
Length[%]
3
Table[Position[%19[[i]],True],{i,1,3}]
{{{2}},{{2}},{{2}}}
```

CURRICULUM VITAE

Personal Information	
Name Surname	Kübra DURAN
Place of Birth	Samsun
Date of Birth	19.07 1991
Nationality	☑ T.C. ❒ Other:
Phone Number	05312938085
Email	k.yucel@windowslive.com
Web Page	

Educational Information	
B. Sc.	
University	Fatih University
Faculty	Faculty of Art And Sience
Department	Mathematics
Graduation Year	01.01.2014

M. Sc.	
University	Istanbul University
Institute	Institute of Graduate Studies in Science and Engineering
Department	Department of Mathematics
Programme	Mathematics Programme
Graduation Year	24.11.2017