AN ABSTRACT OF THE THESIS OF

Der Jei Lin for the degree of Doctor of Philosophy in

Computer Science presented on July 2, 1987.

Title: Unidirectional Error Correcting/Detecting Codes

Redacted for Privacy

Abstract approved:

Bella Bose

An extensive theory of symmetric error control coding has been developed in the last few decades. The recently developed VLSI circuits, ROM, and RAM memories have given an impetus to the extension of error control coding to include asymmetric and unidirectional types of error control.

The maximal numbers of unidirectional errors which can be detected by systematic codes using r checkbits are investigated. They are found for codes with k, the number of information bits, being equal to 2^r and $2^r + 1$. The importance of their characteristic in unidirectional error detection is discussed.

A new method of constructing a systematic t-error correcting/all-unidirectional error detecting(t-EC/AUED) code, which uses fewer checkbits than any of the previous methods, is developed. It is constructed by appending $t + 1$ check symbols to a systematic t-error correcting and (t+1)-error detecting code. Its decoding algorithm is developed. A bound on the number of checkbits for a systematic t-EC/AUED code is also discussed.

Bose-Rao codes, which are the best known single error correcting/all-unidirectional

error detecting(SEC/AUED) codes, are completely analyzed. The maximal Bose-Rao codes for a fixed weight and for all weights are found. Of course, the base group and the group element which make the Bose-Rao code maximal are found, too. The bounds on the size of SEC/AUED codes are discussed.

Nonsystematic single error correcting/d-unidirectional error detecting codes are constructed. Three methods for constructing the systematic t-error correcting/d-unidirectional error detecting(t-EC/d-UED) codes are developed. From these, simple and efficient t-EC/(t+2)-UED codes are derived. The decoding algorithm for one of these methods, which can be applied to the other two methods with slight modification, is described. A lower bound on the number of checkbits for a systematic t-EC/d-UED code is derived.

Finally, future research efforts are proposed.

Unidirectional Error Correcting/Detecting Codes

by

Der Jei Lin

A Thesis

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed July 2, 1987

Commencement June 1988

APPROVED:

Redacted for Privacy

Professor of Computer Science in charge of major

Redacted for Privacy

Chairman of Department of Computer Science

Redacted for Privacy

Dean of Graduate School

Date thesis is presented July 2, 1987

Typed by Der Jei Lin for Der Jei Lin

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Dr. Bella Bose for his many helpful discussions, suggestions, patient guidance, support, and assistance necessary for the completion of this work.

Also, my sincere appreciation to Mr. Russell Ruby for reading the entire manuscript and making numerous suggestions.

I acknowledge the financial support of the Department of Computer Science and the National Science Foundation.

Last, but not least, I wish to thank my entire family, especially my mother Mrs. Su-Er Su Lin and my brother Dr. Der Chia Lin, for their constant encouragement and love. I wish to dedicate this thesis to my beloved mother and to the memory of my father.

TABLE OF CONTENTS

1. Introduction 1

1.1. Coding Problem 1

1.2. Importance of Asymmetric and Unidirectional Error Control Codes for VLSI Technology 3

1.3. Systematicity and Why Systematic Codes? 5

1.4. Definitions, Notations, and Theorems 6

1.5. Outline of the Thesis 9

2. Unidirectional Error Detecting Codes 11

2.1. Introduction 11

2.2. The Basic Theorems of Unidirectional Error Detection 11

2.3. Previous Works in the Literature 12

2.4. Bounds on the Size of a Code and the Number of Errors which can be Detected by Using r Checkbits 14

3. t-Error Correcting and All-Unidirectional Error Detecting Codes 26

3.1. Introduction 26

3.2. The Basic Theorem of t-EC/AUED Code 27

3.3. Previous Works in the Literature 27

3.4. Code Construction 28

3.5. Decoding Algorithm 34

3.6. On the Number of Checkbits and the Size of a Code 41

4. Study of Bose-Rao Codes 45

4.1. Introduction 45

4.2. Some Knowledge from Fourier Analysis 48

4.3. Some Knowledge from Combinatorics 50

4.4. Analysis of Bose-Rao Codes 55

4.5. On the Size of a SEC/AUED Code 72

5. t-Error Correcting and d-Unidirectional Error Detecting Codes 74

5.1. Introduction 74

5.2. The Basic Theorem of t-EC/d-UED Code 75

5.3. Nonsystematic Code Construction for SEC/d-UED Code 75

5.4. Systematic Code Construction for t-EC/d-UED ($1 \leq t$) Code 78

5.5. Decoding Algorithm 90

5.6. On the Number of Checkbits and the Size of a Code 93

6. Conclusion 96

6.1. Summary and Future Research Efforts 96

6.2. Totally Self-Checking Checkers 97

Bibliography 99

LIST OF FIGURES

Figure		Page
1.1	Block diagram of a typical communication or storage system	1
1.2	A binary block code for the communication system in Figure 1.1	2

LIST OF TABLES

Table		Page
3.1	3-EC/AUED code based on C and the Procedure	31
3.2	2-EC/AUED codes	43
3.3	3-EC/AUED codes	43
5.1	The values of d using s bits for CH in SEC/d-UED codes	86
5.2	The values of d using s bits for CH in 2-EC/d-UED codes	86
5.3	The values of d using s bits for CH in 3-EC/d-UED codes	87
5.4	The values of d using s bits for CH in 4-EC/d-UED codes	87
5.5	The values of d using s bits for CH in 5-EC/d-UED codes	87
5.6	The values of d using s bits for CH in 6-EC/d-UED codes	88

Unidirectional Error Correcting/Detecting Codes

Chapter 1

Introduction

1.1. Coding Problem

It is desirable, and in many cases vital, that information remains correct when transmitted from here to there or stored and later recovered. In communication links or computer memories, noise causes the received data to differ slightly from the original data. A block diagram of a typical communication or storage system is shown in Figure 1.1.

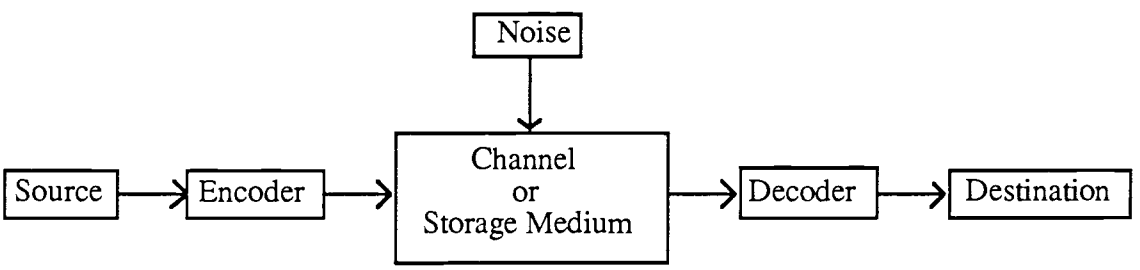

Figure 1.1. Block diagram of a typical communication or storage system.

In addition to the data bits one wishes to transmit, one also transmits some additional redundant checkbits. Even though the noise causes some errors in both the transmitted data bits and the transmitted checkbits, there is usually still enough information available to the receiver to allow a sophisticated decoder to correct or detect the errors unless the noise is extremely severe.

Depending on how the system operates on its input of a continuous sequence of

information digits, there are two fundamentally different types of codes --- a class of block codes and a class of tree codes. Of these two classes of codes, the older block codes have a considerably better developed theory. In the class of tree codes, only the subset called convolutional codes have a substantially developed theory [61]. However, here we consider only block codes. More precisely, we consider only binary block codes, i.e. the source symbols consist of only two possible symbols, which we take to be "0" and "1". A binary block code for the communication system in Figure 1.1 is shown in Figure 1.2,

where u_i , v_i , $i = 1, \cdots, k$, x_j , y_j , $j = 1, \cdots, n$, $\in \{0, 1\}$.

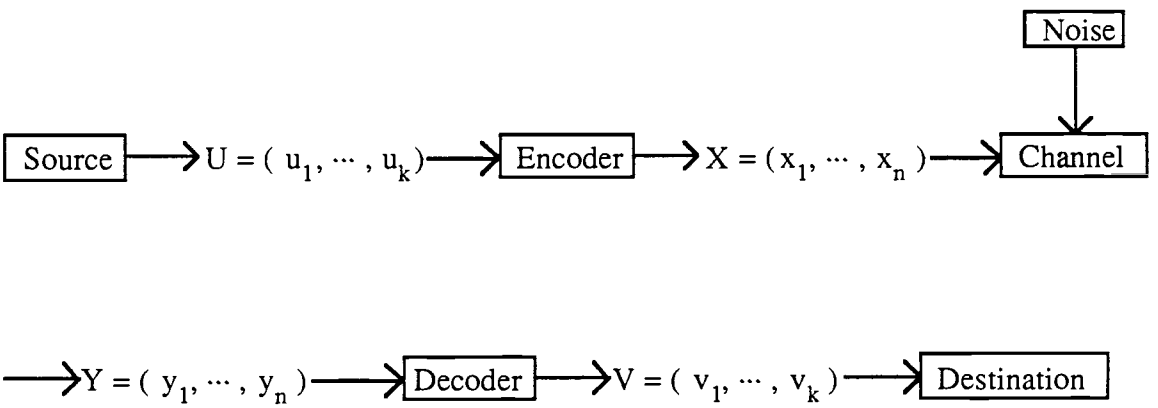

Figure 1.2. A binary block code for the communication system in Figure 1.1.

As suggested by Figure 1.2, the encoder for a binary block code breaks the continuous sequence of information bits into blocks of k bits. It then maps each k-bit source block U independently into an n-bit codeword X, where $k < n$. The n-bit codeword is transmitted over the channel and received, possibly garbled, as Y. The decoder maps each n-bit noisy codeword Y independently into a k-bit block V, which is an estimate of the original source sequence U. The quantity n is referred to as the code length or block length. And the ratio k/n is referred to as the information rate.

The types of error statistics which generated by the noise are many and varied.

However, they can be broadly classified as symmetric, asymmetric, and unidirectional errors. These error classes are defined below [20].

Definition 1.1. If both $0 \rightarrow 1$ errors and $1 \rightarrow 0$ errors occur in a received word with equal probability then the channel is called a binary symmetric channel (the BSC for short) and the errors are called symmetric type.

Definition 1.2. In an ideal asymmetric channel, sometimes called Z-channel, only one type of error can occur and the error type is known a priori. These errors are called asymmetric type.

Definition 1.3. If both $0 \rightarrow 1$ errors and $1 \rightarrow 0$ errors can occur in received words, but in any particular word all errors are of one type, then they are called unidirectional errors.

1.2. Importance of Asymmetric and Unidirectional Error Control Codes for VLSI Technology

Almost four decades have passed since famous mathematician Claude Shannon had published his classic paper "A Mathematical Theory of Communication" in 1948 [69], when he created a completely new branch of applied mathematics which is today called information theory and/or coding theory.

In the first three decades, most of the communication error control codes have been developed under the fault assumption of symmetric errors in the data bits and an extensive theory of symmetric error control coding has been developed, e.g. in [5], [6], [51], [61]. However, in many types of recently developed VLSI circuits the error statistics are different. For example, the failure in the memory cells of some of the VLSI single-

transistor-cell memories and metal-nitride-oxide semiconductor (MNOS) memories are most likely caused by the leakage of charge. If we represent the presence of charge in a cell by 1 and the absence of charge by 0, then the errors in these type of memories can be modelled as $1 \rightarrow 0$ type asymmetric errors. This is because the charge cannot be created except by a rewrite process, and hence $0 \rightarrow 1$ type errors in the memory cells are almost impossible [23], [65].

On the other hand, it is well established that the various faults in many digital devices are the sources of unidirectional errors [1], [24], [31], [32], [52], [60], [64], [81], [82], [84]. Typical digital units which exhibit unidirectional nature of errors caused by their internal failure are: data transmission systems, shift-register and magnetic-recording mass memories, and LSI/VLSI circuits such as ROM memories, PLA's, and interconnection networks. More detailed description of these faults follows.

(1) Typical faults in data transfer systems which cause unidirectional errors are the following:
   - (a) single faults in serial data bus or byte-serial processor (assuming that they are used repeatedly) [82],
   - (b) a failure to enable a register onto a bus [84],
   - (c) bridging faults on a bus or broken bit lines [84].

(2) The faults of shift-register and magnetic-recording mass memories causing undirectional errors are:
   - (a) permanent stuck-at faults in a single register or the malfunctions of the read/write circuitry in shift-register mass memories,
   - (b) the malfunctions of a head due to stuck-at faults of the control circuit or bursts on the magnetic-recording surface due to dust particles, minute scratches, and defects in the coating, both in magnetic-recording mass memories, are assumed in [32], [60], [81], [82] to result in unidirectional

errors.

(3) Very likely sources of unidirectional errors in ROM's are the following [24], [30], [64]:

(a) the faults in address decoder which result in either no access (all 0's word is readout) or multiple access (the OR of several words is readout),

(b) the word line faults such as open line or two word lines shorted together, which beyond the point of the fault cause the same errors as in case 3(a),

(c) power supply faults.

(4) Any of three general types of faults that commonly occur in a PLA, i.e. classical stuck-at fault, the short between two adjacent lines, and the contact fault (it includes the missing device fault and the extra device fault), produce only unidirectional errors at the external outputs of the PLA [30], [52].

(5) Many functional failures in switching elements and failures in the links of the following classes of VLSI implemented interconnection networks: $2^n \times 2^n$ delta networks, centralized control networks with 2×2 crossbar switches, and time-shared buses, also are sources of unidirectional errors [31].

The above specification shows that unidirectional errors are widespread in digital systems and that their correction or detection is a serious problem in the areas of error correcting/detecting codes and fault-tolerant system design.

1.3. Systematicity and Why Systematic Codes?

We start with the well known definition of "systematicity" as follows:

Definition 1.4. Let C be a binary code with codewords of length n, C is a systematic

code if there is a subset of k bits which represent the information bits which are not modified while the remaining n - k bits represent the checkbits. A code which is not systematic is called a nonsystematic code.

The information contained in a codeword in a nonsystematic code cannot be obtained without using a special decoder circuit. Therefore, nonsystematic codes are of restricted use in most computer applications. In contrast, the systematic codes have the advantage that the encoding/decoding and data manipulation can be done in parallel.

1.4. Definitions, Notations, and Theorems

The following special symbols perhaps need explanation: "//" signals the end of a proof or example; "iff" means if and only if; $\lfloor x \rfloor$ denotes the largest integer $\leq x$; and $\lceil x \rceil$ denotes the smallest integer $\geq x$.

Now, let us introduce two fundamental concepts in coding theory [6], [35], [51], [61].

Definition 1.5. Let Q be the set of information symbols.
If $X = (x_1, \cdots, x_n) \in Q^n$, $Y = (y_1, \cdots, y_n) \in Q^n$, then the Hamming distance of X and Y, denoted as $D_H(X, Y)$, is defined by

$$D_H(X, Y) = |\{ i \mid 1 \leq i \leq n, x_i \neq y_i \}|.$$

The Hamming weight of X, denoted as $W_H(X)$, is defined by $W_H(X) = D_H(X, \mathbf{0})$, where $\mathbf{0} = (0, \cdots, 0)$.

In our case, Q = GF(2) = {0, 1}.

Using this concept, Hamming [35] has described the conditions for symmetric error

correcting/detecting codes.

Theorem 1.6. A code C is capable of detecting t or fewer errors iff the minimum Hamming distance of the code is at least $t + 1$.

Theorem 1.7. A code C is capable of correcting t or fewer errors iff the minimum Hamming distance of the code is at least $2t + 1$.

Theorem 1.8. A code C is capable of correcting t or fewer errors and detecting up to d ($d > t$) errors iff the minimum Hamming distance of the code C is at least $t + d + 1$.

As described in Section 1.2, the asymmetric and unidirectional error control codes are becoming important in modern technology. In order to discuss these types of codes, which is the main theme in this thesis, we need some more concepts and notations.

Definition 1.9. If $X = (x_1, \cdots, x_n) \in GF(2)^n$, $Y = (y_1, \cdots, y_n) \in GF(2)^n$, then the number of $1 \rightarrow 0$ crossovers from X to Y, denoted as N(X, Y), is defined by

$$N(X, Y) = |\{ i \mid 1 \le i \le n,\ x_i = 1, y_i = 0 \}|.$$

For example, $N(0011, 1101) = 1$ and $N(1101, 0011) = 2$.

Hamming distance of two binary n-tuples X and Y can be expressed in terms of $1 \rightarrow 0$ crossovers as

$$D_H(X, Y) = N(X, Y) + N(Y, X).$$

Using the parameter N(X, Y), Rao and Chawla [65] and Anderson [1] have defined "asymmetric distance" with reference to the asymmetric error correcting capabilities of binary block codes as follows.

Definition 1.10. If $X = (x_1, \cdots, x_n) \in GF(2)^n$, $Y = (y_1, \cdots, y_n) \in GF(2)^n$, then the asymmetric distance of X and Y, denoted as $D_A(X, Y)$, is defined by

$$D_A(X, Y) = \max(N(X, Y), N(Y, X)).$$

Theorem 1.11. A binary code C is capable of correcting t or fewer asymmetric $1 \rightarrow 0$ errors (or $0 \rightarrow 1$ errors) iff the minimum asymmetric distance of the code is at least $t + 1$.

Note that $D_H(X, Y) = t$ implies $\left\lfloor \frac{t+1}{2} \right\rfloor \leq D_A(X, Y)$. Hence, a code C capable of correcting t symmetric errors must be capable of correcting t asymmetric errors. Therefore, one expects that for a given length n, a t-asymmetric error correcting code will have more codewords (i.e. higher information rate) than a t-symmetric error correcting code. Asymmetric error correcting codes having information rates better than symmetric error correcting codes have been derived in [23], [26], [41], [65], [73], [76], [77], [78], [79]. Also, see [37] which has a clear description of the development of asymmetric error correcting codes. Bose and Cunningham [15] have an interesting discussion on systematic asymmetric error correcting codes, which shows that when $n \neq 2^r$, $2^r + 1$, where n = the code length and r = the number of redundant checkbits, Hamming codes are also optimal systematic asymmetric codes.

Another useful concept is needed when one discusses unidirectional (asymmetric, too) error control codes.

Definition 1.12. A n-tuple $X = (x_1, \cdots, x_n)$ is said to cover another n-tuple $Y = (y_1, \cdots, y_n)$ if $0 < N(X, Y)$ and $N(Y, X) = 0$. Also, they are called an "ordered pair". If $X \neq Y$ and neither X covers Y nor Y covers X, then X and Y are said to be "unordered".

Bose and Rao [20] use this concept and asymmetric distance to describe the unidirectional error correcting capabilities of binary block code.

Theorem 1.13. A code C is capable of correcting t or fewer unidirectional errors iff the following condition holds.

For all distinct $X, Y \in C$,

$2t + 1 \leq D_A(X, Y) = D_H(X, Y)$, if X and Y are ordered pair,

$t + 1 \leq D_A(X, Y)$, otherwise.

By this theorem, a code capable of correcting t symmetric errors is also capable of correcting t unidirectional errors. But, at this point the problem of constructing t-unidirectional error correcting codes having information rates better than that of t-symmetric error correcting codes is still an open research question. Similarly, the problem of constructing t-asymmetric error correcting codes having information rates better than that of t-unidirectional error correcting codes is an open research question, too.

The conditions for asymmetric error detection and the conditions for unidirectional error detection will be discussed in Chapter 2. The conditions for symmetric error correction and unidirectional error detection will be discussed in Chapters 3 and 5.

1.5. Outline of the Thesis

From Chapter 2 through Chapter 5, each topic is treated as an independent topic. The background and the previous work in the literature related to each topic are reviewed in its own chapter. So, we will not do those reviewings here.

In [17] systematic codes are constructed which detect $5 \cdot 2^{r-4} + r - 4$ unidirectional errors by using r checkbits independent of the number of information bits. The question

of the optimality of these codes raises further discussion in Chapter 2. There, we still have no conclusive answer to it. However, we show some facts which tend to indicate the codes might be optimal if k, the number of information bits, is beyond some threshold number.

In Chapter 3, t-error (symmetric error) correcting/all unidirectional error detecting (t-EC/AUED) codes are designed and their decoding algorithms are developed. These codes have been shown to be more efficient than any previous codes. The number of checkbits and the size of a t-EC/AUED code are discussed, too.

Due to the importance of single error correcting/all unidirectional error detecting(SEC/AUED) codes and that Bose-Rao codes are the best of this type so far, even though they are nonsystematic, a complete study of Bose-Rao codes is shown in Chapter 4. There, maximal Bose-Rao codes are found. Also, their sizes are compared with upper bounds derived by Bose.

In Chapter 5, t-error correcting/d-unidirectional error detecting (t-EC/d-UED) codes are designed, which are the first such kind of codes ever designed, and their decoding algorithms are developed. The number of checkbits and the size of a t-EC/d-UED code are discussed, too.

Chapter 6 is the conclusion of this thesis. Some future research topics continuing from this thesis are described.

Chapter 2

Unidirectional Error Detecting Codes

2.1. Introduction

It has been described in Chapter 1 how the recently developed VLSI circuits, ROM, and RAM memories have given an impetus to extension of error control coding to include asymmetric and unidirectional types of error control.

The following notations will be used repeatedly in this chapter.

k : the number of information bits.

r : the number of checkbits.

n : the length of a code.

The basic theorems of unidirectional error detection, which are the principles used in constructing and verifying a code, will be reviewed in Section 2.2. Then all of the previous works related to unidirectional error detection will be reviewed in Section 2.3. The codes in [17] were thought to be near optimal, but afterwards the authors have realized that if k is not much larger than 2^r, then the codes are not optimal (special thanks to Dr. George A. Converse for pointing this out). As a matter of fact, in the recent paper [40] a class of codes is constructed which detects more errors for certain values of k. The various bounds on a unidirectional error detecting code will be discussed in Section 2.4. Basically, this chapter is a supplement of the paper [17].

2.2. The Basic Theorems of Unidirectional Error Detection

The following fundamental theorems describe the necessary and sufficient conditions for an all-unidirectional error detecting code [4], [29] and a t-unidirectional error detecting code [17].

Theorem 2.1. A code C is capable of detecting all unidirectional(asymmetric) errors iff the codewords are unordered (i.e. for all $X, Y \in C$ with $X \neq Y$ implies $1 \leq N(X, Y)$ and $1 \leq N(Y, X)$.)

Theorem 2.2. A code C is capable of detecting t-unidirectional errors iff it satisfies the following condition:

for all $X, Y \in C$ with $X \neq Y$ implies
either $t + 1 \leq D_H(X, Y)$
or $1 \leq N(X, Y)$ and $1 \leq N(Y, X)$.

In [17], it has been shown that the necessary and sufficient condition for a code being capable of detecting t-asymmetric errors is exactly the same as in Theorem 2.2. Thus, there is no difference in constructing and verifying a code being t(or all)-unidirectional error detecting and being t(or all)-asymmetric error detecting.

2.3. Previous Works in the Literature

In this section, all previous code constructions related to unidirectional error detection will be reviewed.

The first such codes were found by Berger [4]. They are systematic all unidirectional errors detecting codes. Freiman [29] gave nonsystematic constant weight

codes to detect all unidirectional errors. However, when not all 2^k information symbols are present, Smith [70] has proposed systematic codes which need fewer check bits than the Berger codes. Berger codes, $\lfloor n/2 \rfloor$-out-of-n codes, and Smith codes are optimal when all unidirectional errors are required to be detected. But when only t-unidirectional errors are required to be detected these codes are not optimal. That is, a better code can be designed. Dong [28] has given modified Berger codes to detect 2^i, $i = 2, 3, 4, \cdots$, unidirectional errors. The number of checkbits needed to detect 2^i unidirectional errors is $i + \lceil \log_2(i + 1) \rceil$. Then, Bose and Lin [17] have constructed systematic t-unidirectional errors detecting codes which require a fixed number of checkbits independent of the number of information bits. Also, the codes presented in [17] have higher error-detecting capabilities than the codes presented in [28].

The codes constructed in [17] are shown to be optimal in the sense of the following two theorems.

Theorem 2.3. Any systematic t-unidirectional error detecting code, where $t = 2, 3$, or 4, requires at least t checkbits.

Theorem 2.4. Any systematic code that detects seven unidirectional errors requires at least five checkbits, if $20 \leq k$.

For $5 \leq r$, the codes constructed in [17] are capable of detecting up to $5 \cdot 2^{r-4} + r - 4$ unidirectional errors. Recently, in [40] a class of systematic t-unidirectional error detecting codes is constructed which detects more than $5 \cdot 2^{r-4} + r - 4$ unidirectional errors when k is not too much larger than 2^r. For longer k, codes in [17] still perform better than codes in [40]. The bounds on the number of errors can be detected by using r checkbits for certain values of k will be discussed in Section 2.4. Although the codes in [40] perform better than the codes in [17] for some

range of k, the latter is much easier to implement(encoding/decoding).

On the other hand, Borden [8] has proved that among all t-unidirectional error detecting codes of length n, the set of codewords with weight $\lfloor n/2 \rfloor \bmod (t+1)$ forms the optimal code.

Two interesting papers [14], [45] require attention, too. Even though the codes in [14], [45] are nonsystematic the encoding/decoding algorithm is simple and easy to implement. First, Knuth [45] has designed balanced codes with 2^r information bits and r checkbits, which need serial decoding. A so-called balanced code is a code with k information bits and r checkbits such that each codeword contains equally many zeros and ones. Naturally, a balanced code is unordered. Therefore, it detects all unidirectional errors. Knuth's coding scheme is interesting in that to construct a systematic unordered code with 2^r information bits requires at least r + 1 checkbits, whereas Knuth's code needs only r checkbits. In [45], a parallel coding scheme for the design of balanced codes with $2^r - r - 1$ information bits and r checkbits is also developed. Bose [14] has extended Knuth's results and designed several efficient unordered codes. They are

(i) parallel unordered coding scheme with 2^r information bits and r checkbits,

(ii) balanced codes with r checkbits and up to $2^r + 2^{r-1} - 1$ information bits which need serial encoding/decoding,

and

(iii) unordered codes with r checkbits and up to $2^r + 2^{r-1} - 1$ information bits which are shown to be optimal.

In [14], a nonsystematic code capable of detecting $2^{r-1} - 1$ unidirectional errors using r checkbits has also been designed.

2.4. Bounds on the Size of a Code and the Number of Errors which can be Detected by Using r Checkbits

Probably the most basic problem in coding theory is to find the most efficient code of given conditions, such as to find the largest code of a given length and minimum distance(or some given error correcting/detecting properties), to find the minimum number of checkbits of a given length of the information and given error correcting/detecting properties in a systematic construction, or to find the maximum number of errors that can be corrected/detected for given lengths of the information and the check, etc..

By studying Berger's paper [4], Freiman's paper [29], and Borden's paper [8], we have already learned the minimum number of checkbits needed for systematic all-unidirectional error detecting code and the upper bounds for an all-unidirectional error detecting code and a t-unidirectional error detecting code. For clarity, their results are restated as the following three theorems.

Theorem 2.5. The minimum number of checkbits needed for a systematic all-unidirectional error detecting code is $\lceil \log_2(k+1) \rceil$.

Theorem 2.6. The number of codewords in an all-unidirectional error detecting code is no more than $\binom{n}{\lfloor \frac{n}{2} \rfloor}$.

(Note: This is also a result from Sperner's Lemma [72].)

Theorem 2.7. The number of codewords in a t-unidirectional error detecting code is no more than $\sum_{w \equiv \lfloor n/2 \rfloor \bmod (t+1)} \binom{n}{w}$.

The codes in [4], [8], and [29] all reach the bounds, i.e. they are optimal. However, the codes in [4] are all-unidirectional error detecting and the codes in [29] and

[8] are nonsystematic. The bounds on systematic t-unidirectional error detecting code still require discussion. Instead of asking for the minimum number of checkbits to detect t errors, one could look for a code using r checkbits which detects the maximal number of errors. Because of Theorems 2.3 and 2.4, here $5 \le r$ is considered. Also, as mentioned in [17], when $k < 2^r$ Berger codes are superior to any systematic t-unidirectional error detecting codes, so only $2^r \le k$ is considered. Before the main theorems are developed, some lemmas are needed.

Lemma 2.8. $2^{\lceil \frac{a}{2} \rceil} + 2^{\lfloor \frac{a}{2} \rfloor} \le 2^b + 2^{a-b}$ for integers a, b, $0 \le b \le a$.

Proof. Let $f(x) = 2^x + 2^{a-x}$ be a real function defined over the interval [0, a]. Then, simple calculus shows that f(x) is decreasing over $[0, \frac{a}{2}]$ and increasing over $[\frac{a}{2}, a]$. Therefore, the inequality holds. //

Lemma 2.9. $2^{a-1} - 2 \le 2^a - 2^b - 2^{a-b}$ for integers a, b, $1 \le b \le a - 1$.

Proof. Let $f(x) = 2^a - 2^x - 2^{a-x} - 2^{a-1} + 2$ be a real function defined over the interval [1, a -1]. Then, simple calculus shows that f(x) is increasing over $[1, \frac{a}{2}]$ and decreasing over $[\frac{a}{2}, a-1]$. And f(1) = f(a -1) = 0. Therefore, the inequality holds. //

Lemma 2.10. $2^a - 2^b - 2^{a-b} \le 2^a - 2^{\lceil \frac{a}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{a}{2} \rfloor - 1} + 1$ for integers a, b, $0 \le b \le a$, and $b \ne \lceil \frac{a}{2} \rceil, \lfloor \frac{a}{2} \rfloor$.

Proof. Let $f(x) = 2^a - 2^{\lceil \frac{a}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{a}{2} \rfloor - 1} + 1 - 2^a + 2^x + 2^{a-x}$ be a real function defined over the interval [0, a]. Then, simple calculus shows that f(x) is decreasing over $[0, \frac{a}{2}]$

and increasing over $[\frac{a}{2}, a]$. And $f(\lfloor\frac{a}{2}\rfloor - 1) = f(\lceil\frac{a}{2}\rceil + 1) > 0$. Therefore, the inequality holds. //

Lemma 2.11. $0 \le 2^{a-1} - 2^{\lfloor\frac{a}{2}\rfloor - 1} - a + 1$ for integer $0 \le a$.

Proof. This lemma will be proved by induction. It is obvious that the inequality holds for $a = 0, 1, 2, 3$. Assuming it holds for $a - 1$, then

$$0 \le 2^{a-2} - 2^{\lfloor\frac{a-1}{2}\rfloor - 1} - (a - 1) + 1 .$$

And,

$$\begin{aligned} 2^{a-1} - 2^{\lfloor\frac{a}{2}\rfloor - 1} - a + 1 &\ge 2^{a-1} - 2^{\lfloor\frac{a-1}{2}\rfloor} - a + 1 \\ &= 2^{a-2} - 2^{\lfloor\frac{a-1}{2}\rfloor - 1} - (a - 1) + 1 + 2^{a-2} - 2^{\lfloor\frac{a-1}{2}\rfloor - 1} - 1 \\ &\ge a - 3 \qquad \text{(by the induction hypothesis)} \\ &\ge 0 . \text{ //} \end{aligned}$$

Theorem 2.12. Let $k = 2^r$. Then, the maximum number of unidirectional errors which can be detected by a systematic code using r checkbits is $2^r - 2^{\lceil\frac{r}{2}\rceil} - 2^{\lfloor\frac{r}{2}\rfloor} + 1$.
In fact, such code exists.

Proof. The proof will be done in two parts. First, a code which detects $2^r - 2^{\lceil\frac{r}{2}\rceil} - 2^{\lfloor\frac{r}{2}\rfloor} + 1$ errors will be constructed. Then, it will be shown that it is optimal.
Let R be a r-tuple vector with $W_H(R) = \lceil\frac{r}{2}\rceil$.
Define

$$M = \{ S \mid S \text{ is a r-tuple vector with } 1 \le N(S, R) \text{ and } 1 \le N(R, S) \},$$

(i.e. M contains those r-tuples which are unordered with R.)

$U = \{ S \mid S$ is a r-tuple vector with $0 < N(S, R)$ and $0 = N(R, S) \}$,

(i.e. U contains those r-tuples which cover R.)

and

$L = \{ S \mid S$ is a r-tuple vector with $0 = N(S, R)$ and $0 < N(R, S) \}$.

(i.e. L contains those r-tuples which are covered by R.)

Note that M, L, U, and {R} form a partition of the set of all 2^r r-tuple vectors.

Also, $|M| = 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1$. Now, arrange the elements in U, M, and L as

$U = \{S_0, S_1, \cdots, S_{|U|-1}\}$, $M = \{S_{|U|+1}, \cdots, S_{|U|+|M|}\}$, and

$L = \{S_{|U|+|M|+2}, \cdots, S_{|U|+|M|+|L|+1}\}$ such that $W_H(S_j) \le W_H(S_i)$

if $0 \le i < j \le |U| - 1$, $|U| + 1 \le i < j \le |U| + |M|$,

or $|U| + |M| + 2 \le i < j \le |U| + |M| + |L| + 1$. (i.e. The elements in each set of U, M and L are arranged in nonincreasing weight order.)

Then, let $C_{|U|} = C_{|U|+|M|+1} = R$ and $C_i = S_i$ for all $0 \le i \le k$ and $i \ne |U|$ and $|U| + |M| + 1$.

e.g. For $r = 5$, let $R = 11100$. Then,

$M = \{11011, 10111, 01111, 11001, 11010, 10101, 10110,$

$01101, 01110, 10011, 01011, 00111, 00101, 01001,$

$10001, 00110, 01010, 10010, 00011, 00001, 00010\}$,

$L = \{11000, 10100, 01100, 10000, 01000, 00100, 00000\}$,

$U = \{11111, 11110, 11101\}$.

Now, assign the check symbol to the information symbol I as $C_{W_H(I)}$.

By Theorem 2.2, it is easy to see that the construction above detects

$2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1$ errors.

Next, this value has to be proved optimal. Let t be the number of errors which can be detected by a code C with $k(=2^r)$ information bits and r checkbits. Let us consider

the check symbols C_0, C_1, $\cdots$, C_k for the information symbols $I_0 = 0 \cdots 0$, $I_1 = 0 \cdots 01$, $I_2 = 0 \cdots 011$, $\cdots$, $I_{k-1} = 011 \cdots 1$, and $I_k = 1 \cdots 1$. Since there are only 2^r different check symbols, it must have some $C_i = C_j$. Assume $i < j$. If $W_H(C_i) = 0$ or $W_H(C_i) = r$, then by Theorem 2.2,

$$t \le r \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1 .$$

For $0 < W_H(C_i) = w < r$, the proof will be discussed in three cases.

Case (i) any C_m, $i < m < j$, is unordered with C_i.

Since the number of symbols which are unordered with C_i is no more than $2^r - 2^w - 2^{r-w} + 1$, by Theorem 2.2 and Lemma 2.8, we have

$$t \le j - i - 1 \le 2^r - 2^w - 2^{r-w} + 1 \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1 . \qquad \text{(by Lemma 2.8)}$$

Case (ii) some C_m, $i < m < j$, covers C_i.

Let q be the smallest integer such that $i < q < j$ and C_q covers C_i.

Also, let p be the largest integer such that $i \le p < q$ and C_q covers C_p.

Then, any C_m, $p < m < q$, is not covered by C_q also does not cover C_i.

And, there are no more than $2^r - 2^w - 2^{r-w} - (2^{w'} - 2^w - 2^{w'-w})$ such C_m's, where $w' = W_H(C_q)$. By Theorem 2.2 and Lemmas 2.9, 2.8, we have

$$\begin{aligned} t &\le q - p - 1 + (w' - 1) \\ &\le 2^r - 2^w - 2^{r-w} - (2^{w'} - 2^w - 2^{w'-w}) + (w' - 1) \\ &\le 2^r - 2^w - 2^{r-w} - 2^{w'-1} + w' + 1 \qquad (2.1) \\ &\le 2^r - 2^w - 2^{r-w} + 1 \\ &\le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1 . \end{aligned}$$

Case (iii) some C_m, $i < m < j$, is covered by C_i.

Similar to the last case, C_p and C_q are chosen such that C_q covers C_p, C_j covers C_p, and any C_m, $p < m < q$, is not covered by C_j also does not cover C_p. Then, the same argument as in the last case shows

$$t \le 2^r - 2^{w'} - 2^{r-w'} - 2^{w-1} + w + 1 \qquad (2.2)$$

where $w' = W_H(C_p)$ and $w' < w$

$$\le 2^r - 2^{w'} - 2^{r-w'} + 1$$

$$\le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1 .$$

Therefore, the assertion holds. //

Theorem 2.13. Let $k = 2^r + 1$. Then, the maximum number of unidirectional errors which can be detected by a systematic code using r checkbits is $2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2$. In fact, such code exists.

Proof. The proof consists two parts: the existence of the code and the optimality of the code.

The Existence.

Let X, Y be two r-tuple vectors with $W_H(X) = \lceil \frac{r}{2} \rceil$ and $W_H(Y) = \lfloor \frac{r}{2} \rfloor$, also $N(X, Y) = \lceil \frac{r}{2} \rceil - \lfloor \frac{r}{2} \rfloor + 1$ and $N(Y, X) = 1$.

Define

$B_1 = \{S \mid S \text{ covers } X\}$,

$B_2 = \{S \mid S \text{ covers } Y \text{ but not } X\}$,

$B_3 = \{S \mid S \text{ is unordered with } X \text{ and } Y\}$,

$B_4 = \{S \mid S \text{ is covered by } X \text{ but not } Y\}$,

and

$B_5 = \{S \mid S \text{ is covered by } Y\}$.

The elements in each of the sets above are ordered so that the weights of the elements in each set are nonincreasing. Then $C_0, C_1, \cdots, C_k$ are defined as in the following order.

$$B_1, X, B_2, Y, B_3, X, B_4, Y, B_5.$$

e.g. For $r = 5$, let $X = 11100$, $Y = 10010$. Then

$B_1 = \{11111, 11110, 11101\}$,

$B_2 = \{11011, 10111, 11010, 10110, 10011\}$,

$B_3 = \{01111, 00111, 01101, 01110, 01011, 10101, 11001,$
$00101, 01001, 10001, 00110, 01010, 00011, 00001\}$,

$B_4 = \{11000, 10100, 01100, 01000, 00100\}$,

$B_5 = \{10000, 00010, 00000\}$,

$C_1 = 11110$, $C_3 = 11100$, $C_6 = 11010$, $C_9 = 10010$, $C_{30} = 10010$, etc..

Now, assign the check symbol to the information symbol I as $C_{W_H(I)}$.

By Theorem 2.2, it is easy to see that the construction above detects

$\min\{|B_2| + |B_3| + 1, |B_3| + |B_4| + 1\}$ errors.

Since

$$|B_2| = |B_4| = 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} - 1 \quad \text{and} \quad |B_3| = 2^r - 2^{\lceil \frac{r}{2} \rceil + 1} - 2^{\lfloor \frac{r}{2} \rfloor} + 2,$$

thus, $\min\{|B_2| + |B_3| + 1, |B_3| + |B_4| + 1\} = 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor} + 2$.

Therefore, such code exists.

<u>The Optimality.</u>

Let t be the number of errors can be detected by a code C with $k(= 2^r + 1)$ information bits and r checkbits. Let us consider the check symbols $C_0, C_1, \cdots, C_k$ for the information symbols $I_0 = 0 \cdots 0$, $I_1 = 0 \cdots 01$, $I_2 = 0 \cdots 011$, $\cdots$, $I_{k-1} = 011 \cdots 1$, and $I_k = 1 \cdots 1$. Since there are only 2^r different check symbols, either some symbol is used at least three times or at least two symbols are used twice.

Case (i) some symbol is used at least three times.

Say $C_f = C_g = C_h$, $0 \le f < g < h \le k = 2^r + 1$. By Theorem 2.2, it is easy to see that

$$t \le \min\{g - f - 1,\ h - g - 1\} \le \frac{2^r - 1}{2} \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 \ .$$

Case (ii) at least two symbols are used twice.

Say $C_f = C_i$ and $C_g = C_j$. May assume $f < g$, $f < i$, and $g < j$.

Then, we have these possibilities: $f < i < g < j$, $f < g < j < i$, and $f < g < i < j$.

The proof on this case will be discussed in three subcases according to these possibilities.

Subcase 1 $f < i < g < j$.

This is an easy case. By Theorem 2.2, we have

$$t \le \min\{i - f - 1,\ j - g - 1\} \le \frac{2^r - 2}{2} \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 \ .$$

Before the discussion of the next two subcases, trimming will be done here to shorten the discussion.

First of all, if $W_H(C_f)$ (or $W_H(C_g)$) $= 0$ or r, then no matter which case, by Theorem 2.2,

$$t \le r \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 \ .$$

So, assume $0 < W_H(C_f), W_H(C_g) < r$.

Next, if some C_m, $f < m < i$ (or $g < m < j$) , is ordered with C_f (or C_g) then by the proof of Theorem 2.12 and Lemmas 2.8, 2.11, we have

$$t \le 2^r - 2^w - 2^{r-w} - 2^{r-1} + r + 1 \quad \text{for some } 0 \le w \le r$$

(see (2. 1) and (2. 2))

$$\le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} - 2^{r-1} + r + 1 \qquad \text{(by Lemma 2.8)}$$

$$\leq 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 . \qquad \text{(by Lemma 2.11)}$$

Last, in either case, even if any C_m, $f < m < i$ (or $g < m < j$), is unordered with C_f (or C_g) but if $W_H(C_f)$ (or $W_H(C_g)$) $\neq \lceil \frac{r}{2} \rceil$, $\lfloor \frac{r}{2} \rfloor$, then by the proof of Theorem 2.12(Case (i)) and Lemma 2.10, we have

$$t \leq 2^r - 2^w - 2^{r-w} + 1 \qquad \text{where } w \neq \lceil \tfrac{r}{2} \rceil , \ \lfloor \tfrac{r}{2} \rfloor$$

$$\leq 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 . \qquad \text{(by Lemma 2.10)}$$

Therefore, only when $W_H(C_f) = \lceil \frac{r}{2} \rceil$ or $\lfloor \frac{r}{2} \rfloor$, $W_H(C_g) = \lceil \frac{r}{2} \rceil$ or $\lfloor \frac{r}{2} \rfloor$, any C_m, $f < m < i$, is unordered with C_f, and any C_m, $g < m < j$, is unordered with C_g, further discussion is needed. So, the following discussions on Subcases 2 and 3 are based on these conditioins.

Let $W_H(C_f) = w_1$, $W_H(C_g) = w_2$, $N(C_f, C_g) = v_1$, and $N(C_g, C_f) = v_2$. Since C_f has to be unordered with C_g for either case, so $1 \leq v_1$ and $1 \leq v_2$.

<u>Subcase 2</u> $f < g < j < i$.

Any C_m, $g < m < j$, has to be unordered with C_f and C_g. There are no more than $2^r - 2^{w_1} - 2^{r-w_1} + 1 - 2^{w_2} + 2^{w_1 - v_1} - 2^{r-w_2} + 2^{r-w_1-v_2}$ (called S) such C_m's. And,

$$S = 2^r - 2^{\lceil \frac{r}{2} \rceil + 1} - 2^{\lfloor \frac{r}{2} \rfloor + 1} + 2^{w_1 - v_1} + 2^{r-w_1-v_2} + 1$$

$$= (2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2)$$

$$- (2^{\lceil \frac{r}{2} \rceil} + 2^{\lfloor \frac{r}{2} \rfloor - 1} - 2^{w_1 - v_1} - 2^{r-w_1-v_2} + 1)$$

$$\leq 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 .$$

Therefore,

$$t \le j - g - 1 \le S \le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 .$$

Subcase 3 $f < g < i < j$.

Let S be the number of symbols which are unordered with C_f and are not covered by C_g.

Then,

$$S = 2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1 - 2^{w_2} + 2^{w_1 - v_1} .$$

If some C_m, $f < m < g$, is covered by C_g, then let p be the largest such m. Also, let q be the smallest integer such that $p < q \le g$ and C_q covers C_p. Thus, any C_m, $p < m < q$, must be unordered with C_f and not be covered by C_g, also does not cover C_p.

And, there are at most $S - 2^{r - w_2} + 2^{r - w_2 - v_1}$ such C_m's.

Hence,

$$\begin{aligned} t &\le q - p - 1 + r - 1 \\ &\le S - 2^{r - w_2} + 2^{r - w_2 - v_1} + r - 1 \\ &= (2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2) - (2^{\lceil \frac{r}{2} \rceil} + 2^{\lfloor \frac{r}{2} \rfloor} - 2^{w_1 - v_1} - 2^{r - w_2 - v_1} - r) \\ &\le 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 . \end{aligned}$$

If no C_m, $f < m < g$, is covered by C_g, then any C_m, $f < m < i$, must be unordered with C_f and not be covered by C_g.

Thus,

$$\begin{aligned} t &\le S \\ &= (2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2) + (2^{\lfloor \frac{r}{2} \rfloor - 1} - 2^{w_2} + 2^{w_1 - v_1} - 1) \end{aligned}$$

$$\leq 2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2 .$$

Therefore, the assertion holds. //

Theorems 2.12 and 2.13 still depict very little about the bounds on the number of unidirectional errors can be detected by using r checkbits. But, the phenomenon of the decrease of t, the number of errors which can be detected by a code using r checkbits, (if one tries to extend Theorems 2.12 and 2.13 to $k = 2^r + 2$, one may find that $t = 2^r - 2^{\lceil \frac{r}{2} \rceil + 1} - 2^{\lfloor \frac{r}{2} \rfloor - 2} + 1$, if r = odd, $2^r - 2^{\frac{r}{2}+2} + 2^{\frac{r}{2}} + 1$, if r = even), makes us ponder again whether the codes constructed in [17] may be optimal if k is large enough. Even the codes constructed in [40] show this phenomenon, too. Last, a note about the codes in [40] and Theorems 2.12 and 2.13. The codes in [40] are not optimal. For instance, when $k = 33$ and $r = 5$, the code in [40] detects 18 errors but the code in Theorem 2.13 detects 20 errors; also when $k = 2048$ and $r = 11$ the code in [40] detects 1949 errors but the code in Theorem 2.12 detects 1953 errors.

Chapter 3

t-Error Correcting and All-Unidirectional Error Detecting Codes

3.1. Introduction

Error correcting/detecting codes that are effective against both symmetric and unidirectional errors are useful in providing protection against transient, intermittent, and permanent faults [19], [21], [58], [59], [62]-[64], [81]. Transient faults are likely to cause a limited number of symmetric errors or multiple unidirectional errors [62], [64]. Also, intermittent faults, because of short duration [74], are expected to cause limited number of errors. On the other hand, permanent faults cause either symmetric or unidirectional errors, depending on the nature of the faults [32], [60], [64]. The most likely faults in some of the recently developed LSI/VLSI, ROM, and RAM memories, such as faults that affect address decoders, word lines, power supply, and stuck-fault in a serial bus, etc. [19], [24], [60], [64], cause unidirectional errors. The number of symmetric errors is usually limited while the number of unidirectional errors, caused by the above mentioned faults, can be fairly large. Therefore, people are interested in designing codes which correct all patterns of t or fewer symmetric errors and detect all (t + 1 or more) unidirectional errors [10], [11], [19], [20], [59], [75].

In this chapter, the basic theorem of a t-error correcting/all unidirectional error detecting(t-EC/AUED) code is reviewed (see Section 3.2). Some previous code constructions, including nonsystematic codes and systematic codes, are also reviewed (see Section 3.3). Then improved systematic codes are proposed (see Sections 3.4, 3.5, and 3.6). Further, bounds on the size of a code and on the number of redundant bits for the

systematic code are investigated (see Section 3.6).

3.2. The Basic Theorem of t-EC/AUED Code

Using the parameter N(X, Y) introduced in Chapter 1, a t-EC/AUED code can be characterized as below [9], [19]-[21], [62], [63].

Theorem 3.1. A code C is t-EC/AUED iff it satisfies the following condition:

for all $X, Y \in C$ with $X \neq Y$ implies $t + 1 \leq N(X, Y)$ and $t + 1 \leq N(Y, X)$.

Among all t-EC/AUED codes the single error correcting/all unidirectional error detecting(SEC/AUED) codes are particularly popular. In fact, some feel that in the future SEC/AUED codes may become as popular as today's distance four Hamming codes, which now dominate the applications in computer memories [10], [39]. For this sake the special case, when $t = 1$, of Theorem 3.1 is restated as a corollary.

Corollary 3.2. A code C is SEC/AUED iff it satisfies the following condition:

for all $X, Y \in C$ with $X \neq Y$ implies $2 \leq N(X, Y)$ and $2 \leq N(Y, X)$.

3.3. Previous Works in the Literature

Nonsystematic SEC/AUED codes have been constructed in [9], [20], [63]. For the codes constructed in [63] and the codes constructed by the first method in [9], [20] no efficient encoding/decoding algorithm have been found. Also, the encoding/decoding

algorithm devised in [9] for the codes constructed by the second method in [9], [20] is efficient only for very short information lengths. Nonsystematic t-EC/AUED codes with low redundancy have been constructed in [58].

Although the codes constructed by the first method in [9], [20] are nonsystematic and have no efficient encoding/decoding algorithm, their sizes, being close to the upper bounds, require particular attention. These codes will be analyzed in Chapter 4.

Systematic SEC/AUED codes have been given in [9], [10]. In [62] Pradhan has devised systematic t-EC/AUED codes. More efficient systematic t-EC/AUED codes have been constructed in [19] for moderate and long information lengths. Recently, in [59], [75] new methods for the construction of systematic t-EC/AUED codes, which for most cases are more efficient than the methods used in [9], [10], [19], [62], have been given.

In the next section, a method for the construction of systematic t-EC/AUED codes which is more efficient than the methods used in [59], [75] will be proposed.

3.4. Code Construction

To construct a systematic t-EC/AUED code, $t + 1$ check symbols are appended to a t-error correcting and (t+1)-error detecting systematic parity check code.(e.g. a linear systematic (n, k) code with Hamming distance $2t + 2$. Refer to [6], [51], or [61] for this type of codes.) One may add an even parity bit to a systematic parity check code with Hamming distance $2t + 1$ to make a systematic parity check code with Hamming distance $2t + 2$. In the sequel, let C represent a linear systematic (n, k) code with $D_H(C) = 2t + 2$, and C^* be the systematic t-EC/AUED code constructed from C. Then, any $X^* \in C^*$ has the following form:

$$X^* = X\,CH_1^X \cdots CH_{t+1}^X,$$

where $X \in C$ and CH_i^X, $i = 1, \cdots, t+1$, are the added check symbols.

Before describing how to construct CH_i, $i = 1, \cdots, t+1$, let us partition C into $n+1$ parts, $\{C_0, C_1, \cdots, C_n\}$, where $C_i = \{X \in C \mid W_H(X) = i\}$ and n = the code length of C. Note that some of the C_i's may be empty. Now for the construction.

Procedure (for generating CH_i, $i = 1, \cdots, t+1$)

Step 1.

For $i := 1$ to $t+1$ do

begin

partition the collection $\{C_0, C_1, \cdots, C_n\}$ into $k(i)+1$ subcollections, called blocks $B_{i\,j}$, $j = 0, 1, \cdots, k(i)$, where

$$k(i) = \left\lceil \frac{(n+1) - (2t+2) + 2(i-1)}{2t+2} \right\rceil,$$

as

$B_{i\,k(i)} = \{C_0, C_1, \cdots, C_{2t-2i+3}\}$

containing $(2t+2) - 2(i-1)$ consecutive C_i's starting C_0,

$B_{i\,(k(i)-1)} = \{C_{2t-2i+4}, \cdots, C_{4t-2i+5}\}$

containing the next $2t+2$ consecutive C_i's,

$B_{i\,(k(i)-2)} = \{C_{4t-2i+6}, \cdots, C_{6t-2i+7}\}$

containing the next $2t+2$ consecutive C_i's,

and so on till $B_{i\,1}$,

(i.e. Each of $B_{i\,j}$, $j = 1, \cdots, k(i)-1$, is of size $2t+2$ and contains the first $2t+2$ consecutive C_i's of $\{C_0, \cdots, C_n\} - \bigcup_{m=j+1}^{k(i)} B_{i\,m}$.)

and

$$B_{i\,0} = \{C_0, \cdots, C_n\} - \bigcup_{j=1}^{k(i)} B_{i\,j}$$

(i.e. $B_{i\,0}$ contains the rest of C_i's.)

end.

Step 2.

For $X \in C$, $X \in C_w$ for some w. Then CH_i^X is the binary representation of j (in $\lceil \log_2 (k(i) + 1) \rceil$ bits), where $C_w \in B_{i\,j}$.

End(of Procedure).

Before showing that the proposed code C* is t-EC/AUED, an example is shown here to illustrate the proposed code construction technique.

Example 3.1. A systematic 3-EC/AUED code with three information bits will be constructed. Let C be the 3-error correcting and 4-error detecting code with generator matrix

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

Then, $n = 15$, $D_H(C) = 8$, $k(1) = 1$, $k(2) = 2$, $k(3) = 2$, and $k(4) = 2$. Also,

$$\begin{aligned} &B_{1\,1} = \{C_0, \cdots, C_7\}, \quad B_{1\,0} = \{C_8, \cdots, C_{15}\}; \\ &B_{2\,2} = \{C_0, \cdots, C_5\}, \quad B_{2\,1} = \{C_6, \cdots, C_{13}\}, \quad B_{2\,0} = \{C_{14}, C_{15}\}; \\ &B_{3\,2} = \{C_0, \cdots, C_3\}, \quad B_{3\,1} = \{C_4, \cdots, C_{11}\}, \quad B_{3\,0} = \{C_{12}, \cdots, C_{15}\}; \\ &B_{4\,2} = \{C_0, C_1\}, \qquad B_{4\,1} = \{C_2, \cdots, C_9\}, \qquad B_{4\,0} = \{C_{10}, \cdots, C_{15}\}. \end{aligned} \tag{3.1}$$

Table 3.1 presents all the codewords of the 3-EC/AUED code constructed from the above parity check code C, according to the proposed technique. Column X contains the codewords from C, each of which is presented as three information bits followed by twelve check bits. //

Table 3.1. 3-EC/AUED code based on C and the Procedure.

X		CH_1	CH_2	CH_3	CH_4
000	000000000000	1	10	10	10
001	111100110001	0	01	01	01
010	000011111101	0	01	01	01
011	111111001100	0	01	01	00
100	110011000111	0	01	01	01
101	001111110110	0	01	01	00
110	110000111010	0	01	01	01
111	001100001011	0	01	01	01

Before proving that the proposed codes are indeed t-EC/AUED codes, we prove the following two lemmas which give some properties of distance $2t + 2$ code. These properties are useful in proving the error correcting and detecting capabilities of the codes designed in this chapter and the codes will be designed in Chapter 5.

Lemma 3.3. Let C be a distance $2t + 2$ code with length n. Then there exists a code C' with $|C'| = |C|$, same minimum distance and length. In addition, all the codewords in C' will have even number of 1's.

Proof. Delete the least significant bit of C. Then the resultant codewords will have minimum distance at least $2t + 1$. Now append an even parity bit to these codewords. The

resultant codewords, C', will have even number of 1's and minimum distance $2t + 2$ with $|C'| = |C|$. //

Thus we can always assume that the weight of each codeword in a distance $2t + 2$ code is even.

Lemma 3.4. Let C be a code with $D_H(C) = 2t + 2$. For any $X, Y \in C$, let $q = W_H(X) - W_H(Y)$. (Without loss of generality we assume that $0 \le q$.) Note that q is an even number.

(i) If $2t + 2 \le q$ then $q \le N(X, Y)$ and $0 \le N(Y, X)$.

(ii) If $0 \le q \le 2t$ then $\frac{2t + 2 + q}{2} \le N(X, Y)$ and $\frac{2t + 2 - q}{2} \le N(Y, X)$.

Proof. Since $0 \le q$, there exists q positions where X has 1's and Y has 0's. X and Y will have the same number of 1's in the remaining $n - q$ positions. Thus, the numbers of crossovers from X to Y and from Y to X are equal in these remaining $n - q$ positions. That is,

$$N(X, Y) = q + N(Y, X). \tag{3. 2}$$

Also, recall that $D_H(C) = 2t + 2$. Therefore, by (3. 2), we have

$$2t + 2 \le D_H(X, Y) = N(X, Y) + N(Y, X) = q + 2N(Y, X). \tag{3. 3}$$

(i) If $2t + 2 \le q$, it is obvious that $q \le N(X, Y)$ and $0 \le N(Y, X)$.

(ii) If $0 \le q \le 2t$, by (3. 3), we have $2t + 2 - q \le 2N(Y, X)$, or equivalently

$$\frac{2t + 2 - q}{2} \le N(Y, X).$$

And hence, by (3. 2), we also have

$$\frac{2t + 2 + q}{2} = \frac{2t + 2 - q}{2} + q \le N(Y, X) + q = N(X, Y). \text{ //}$$

Now, let us prove that the proposed codes are indeed t- EC/AUED codes.

Theorem 3.5. The proposed code C^* is t-EC/AUED.

Proof. Let $X^*, Y^* \in C^*$, $X^* = XCH_1^X \cdots CH_{t+1}^X$, and $Y^* = YCH_1^Y \cdots CH_{t+1}^Y$,

where $X, Y \in C$. According to Theorem 3.1, it needs to be shown that

$t + 1 \le N(X^*, Y^*)$ and $t + 1 \le N(Y^*, X^*)$. Without loss of generality, assume

$W_H(Y) \le W_H(X)$. And, let $q = W_H(X) - W_H(Y)$.

The theorem will be proved by discussing the following two cases.

Case (i) $2t + 2 \le q$.

By (i) of Lemma 3.4, $t + 1 < q \le N(X, Y)$.

Hence, $t + 1 < N(X, Y) \le N(X^*, Y^*)$.

Since $2t + 2 \le q$, $C_{W_H(X)}$ and $C_{W_H(Y)}$ will never be in the same block in Step 1 of the Procedure. Thus,

$$CH_i^X < CH_i^Y \quad \text{for all } i = 1, \cdots, t + 1.$$

Therefore,

$$t + 1 \le \sum_{i=1}^{t+1} N(CH_i^Y, CH_i^X) \le N(Y^*, X^*).$$

Case (ii) $0 \le q \le 2t$.

By (ii) of Lemma 3.4, $t + 1 \le N(X, Y) \le N(X^*, Y^*)$. Since $0 \le q \le 2t$, $C_{W_H(X)}$ and $C_{W_H(Y)}$ can be in the same block in Step 1 of the Procedure at most $\frac{2t + 2 - q}{2}$ times. Thus, there are at least $\frac{q}{2}$ i's in which $CH_i^X < CH_i^Y$.

Therefore, by this reason and (ii) of Lemma 3.4, we have

$$t + 1 = \frac{2t + 2 - q}{2} + \frac{q}{2}$$

$$\leq N(Y, X) + \sum_{i=1}^{t+1} N(CH_i^Y, CH_i^X)$$

$$= N(Y^*, X^*). \ //$$

3.5. Decoding Algorithm

In this section a decoding algorithm for error correction and detection of the code described in Section 3.4 is developed. Let $X^* = XCH_1^X \cdots CH_{t+1}^X$ be an error free transmitted codeword in the proposed t-EC/AUED code and $(X^*)' = X'(CH_1^X)' \cdots (CH_{t+1}^X)'$ be the received word with some errors in X^*.

Decoding Algorithm

(1) Compute syndrome S of X' as usual in code C. (May refer to any coding theory book listed in the bibliography, such as [48], [56], or [61], etc..)

Let m be the multiplicity of errors corresponding to the syndrome S.

(2) Compute $CH_i^{X'}$, $i = 1, \cdots, t + 1$, for X' using the Procedure described in Section 3.4.

(3) If $m = 0$ and $D_H((CH_1^X)' \cdots (CH_{t+1}^X)', \ CH_1^{X'} \cdots CH_{t+1}^{X'}) = 0$, then output the codeword $X'(CH_1^X)' \cdots (CH_{t+1}^X)'$ and stop.

(4) If $t < m$ then signal "errors detected" and stop.

(5) Decode X' using a decoding algorithm in code C to get X".

Compute $CH_i^{X"}$, $i = 1, \cdots, t + 1$, for X".

If $m + D_H((CH_1^X)' \cdots (CH_{t+1}^X)', \ CH_1^{X"} \cdots CH_{t+1}^{X"}) \leq t$, then

output the codeword $X"CH_1^{X"} \cdots CH_{t+1}^{X"}$ and stop,

else signal "errors detected" and stop.

End(of Decoding Algorithm).

Notice that steps (1) and (2) of the above algorithm can run in parallel. Before proving the validity of the algorithm, an example is shown here to illustrate how the algorithm works.

Example 3.2. Let us consider the 3-EC/AUED code of Example 3.1 in Section 3.4. The parity check matrix which corresponds to the generator matrix G is

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

Let $X^* = 101001111110110 \quad 0 \quad 01 \quad 01 \quad 00$

$$X \qquad\qquad CH_1^X \; CH_2^X \; CH_3^X \; CH_4^X \; .$$

Suppose that three random errors have occurred in the above codeword when transmitted, in the positions marked with e, shown below. Let the received word be

$$\begin{array}{l} \qquad\quad e \qquad\quad e \qquad\qquad\qquad\qquad\qquad\quad e \\ (X^*)' = 111001111100110 \qquad 0 \qquad\quad 01 \qquad\quad 00 \qquad\quad 00 \\ \qquad\qquad X' \qquad\qquad\quad (CH_1^X)' \;\; (CH_2^X)' \;\; (CH_3^X)' \;\; (CH_4^X)' \; . \end{array}$$

The decoding.

(1) Compute syndrome S of X'. $S = H(X')^T = [0\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0]^T$.

Since S is equal to the sum of the second and the eleventh columns of the parity check matrix H, it indicates that two errors have occurred. Thus, m = 2.

(2) Using (3. 1) of Example 3.1, compute $CH_i^{X'}$, i = 1, ··· , t + 1.

$CH_1^{X'}$	$CH_2^{X'}$	$CH_3^{X'}$	$CH_4^{X'}$
0	01	01	00

Steps (3) and (4) are skipped, since conditions are false.

(5) Decode X' to get X" = 101001111110110.

Using (3. 1) of Example 3.1, compute $CH_i^{X''}$, i = 1, ··· , t + 1.

$CH_1^{X''}$	$CH_2^{X''}$	$CH_3^{X''}$	$CH_4^{X''}$
0	01	01	00

Then, $m + D_H(0010000, 0010100) = 3 \leq 3$. Thus, the Algorithm outputs 10100111111011000010100, which is the correct codeword X*.

Next, let us consider the occurrence of six unidirectional errors in the same codeword X*. Let the received word be

	eeee e			e	
(X*)' = 101001100000010	0	01		00	00
X'	$(CH_1^X)'$	$(CH_2^X)'$		$(CH_3^X)'$	$(CH_4^X)'$.

The decoding.

(1) Compute syndrome S of X'. $S = H(X')^T = [1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0]^T$.

S indicates that three errors have occurred (in second, twelfth, and fifteenth bits). Thus, m = 3.

(2) Using (3. 1) of Example 3.1, compute $CH_i^{X'}$, i = 1, ··· , t + 1.

$CH_1^{X'}$	$CH_2^{X'}$	$CH_3^{X'}$	$CH_4^{X'}$
1	10	01	01

Steps (3) and (4) are skipped.

(5) Decode X' to get $X'' = 111001100001011$. Compute $CH_i^{X''}$, $i = 1, \cdots, t+1$.

$CH_1^{X''}$	$CH_2^{X''}$	$CH_3^{X''}$	$CH_4^{X''}$
0	01	01	01

Then, $m + D_H(0010000, 0010101) = 5 > 3$. Thus, the Algorithm signals "errors detected". //

The validity of the Decoding Algorithm will be proved in the next theorem.

Theorem 3.6. The Decoding Algorithm described above is valid.

Proof. To prove the validity of the algorithm, it needs to be shown that

(i) if t or fewer errors have occurred in the received word, the algorithm outputs the correct codeword,

and

(ii) if $t+1$ or more unidirectional errors have occurred in the received word, the algorithm should signal "errors detected".

Let m_1 and m_2 be the numbers of errors that have occurred in X and $(CH_1^X \cdots CH_{t+1}^X)$, respectively.

Case (i) t or fewer errors.

It is obvious that if $m_1 + m_2 = 0$ the algorithm outputs the correct codeword at step (3). Now, let us consider the situation $1 \leq m_1 + m_2 \leq t$.

Naturally, $m_1 \leq t$, therefore, $m = m_1$ at step (1). It is obvious that the algorithm skips steps (3) and (4). And, at step (5), by the structure of C, $X'' = X$.

Hence, $CH_1^{X''} \cdots CH_{t+1}^{X''} = CH_1^X \cdots CH_{t+1}^X$. Thus,

$$m + D_H((CH_1^X)' \cdots (CH_{t+1}^X)',\ CH_1^{X''} \cdots CH_{t+1}^{X''}) = m_1 + m_2 \le t .$$

Therefore, the algorithm outputs the correct codeword

$$X''CH_1^{X''} \cdots CH_{t+1}^{X''} = XCH_1^X \cdots CH_{t+1}^X .$$

Case (ii) $t + 1$ or more unidirectional errors.

If $t < m$, then step (4) does the job. Thus, it needs only to discuss the situation $m \le t$. And, the discussion will be divided into two subcases.

Subcase (1) $m_1 \le t$.

Since C is t-error correcting and (t+1)-error detecting code, we have $m = m_1$.

If $m_1 = 0$, then $X = X'$ and $t + 1 \le m_2$. Hence, the condition in step (3) will never happen. Also, since $m = m_1 \le t$, step (4) is skipped, too. Thus, whether $m = 0$ or not, the algorithm always executes step (5).

It gets $X'' = X$. Hence, $CH_1^{X''} \cdots CH_{t+1}^{X''} = CH_1^X \cdots CH_{t+1}^X$. Then,

$$t + 1 \le m_1 + m_2 = m + D_H((CH_1^X)' \cdots (CH_{t+1}^X)',\ CH_1^{X''} \cdots CH_{t+1}^{X''}) .$$

Therefore, the algorithm signals "errors detected".

Subcase (2) $t < m_1$.

First of all, let us notice that we have the following fact:

(i) $CH_i^X \le (CH_i^X)'$ for all $i = 1, \cdots, t + 1$, if $0 \to 1$ error occurred,

or (3. 4)

(ii) $(CH_i^X)' \le CH_i^X$ for all $i = 1, \cdots, t + 1$, if $1 \to 0$ error occurred.

This subcase will be proved by discussing the following two subsubcases

(a) $m = 0$ and (b) $0 < m$.

Subsubcase (a) $m = 0$.

If $m = 0$, then $X' = X + Y$ with $0 \neq Y \in C$. Since the error is unidirectional, we have

$$W_H(X') - W_H(X) = \begin{cases} W_H(Y), & \text{if } 0 \rightarrow 1 \text{ error occurred,} \\ \\ -W_H(Y), & \text{if } 1 \rightarrow 0 \text{ error occurred.} \end{cases}$$

According to the code structure of C, $2t + 2 \leq W_H(Y)$, thus,

$$\text{(i) } CH_i^{X'} < CH_i^X \text{ for all } i = 1, \cdots, t+1, \text{ if } 0 \rightarrow 1 \text{ error occurred,}$$

or (3. 5)

$$\text{(ii) } CH_i^X < CH_i^{X'} \text{ for all } i = 1, \cdots, t+1, \text{ if } 1 \rightarrow 0 \text{ error occurred.}$$

Combining (3. 4) and (3. 5), we have

$$0 < t+1 \leq D_H((CH_1^X)' \cdots (CH_{t+1}^X)', \ CH_1^{X'} \cdots CH_{t+1}^{X'}) . \tag{3. 6}$$

Therefore, the algorithm skips step (3). Since $0 = m \leq t$, the algorithm skips step (4), too. In step (5), the algorithm computes $X'' = X'$, because of $m = 0$. Thus, by (3. 6), we have

$$t < m + D_H((CH_1^X)' \cdots (CH_{t+1}^X)', \ CH_1^{X''} \cdots CH_{t+1}^{X''}) .$$

Therefore, the algorithm signals "errors detected".

Subsubcase (b) $0 < m$.

Recall that $m \leq t$. Then, the algorithm skips steps (3) and (4) and executes step (5). It decodes X' as $X'' = X' + A$ with $W_H(A) = m$. Since the error from X

to X' is unidirectional, we have $X' = X + B$ with $W_H(B) = m_1$, and

$$W_H(X') - W_H(X) = \begin{cases} m_1, & \text{if } 0 \rightarrow 1 \text{ error occurred,} \\ -m_1, & \text{if } 1 \rightarrow 0 \text{ error occurred.} \end{cases} \tag{3. 7}$$

Note that $A + B = X + X'' \in C$. Hence,

$$2t + 2 \leq m + m_1 . \tag{3. 8}$$

Now, consider this

$$W_H(X'') - W_H(X) = W_H(X' + A) - W_H(X) = W_H(X') - W_H(X) \pm d,$$
$$\text{where } 0 \leq d \leq m. \tag{3. 9}$$

Combining (3. 7) and (3. 9), we have

$$W_H(X'') - W_H(X) = \begin{cases} m_1 \pm d, & \text{if } 0 \rightarrow 1 \text{ error occurred,} \\ -m_1 \pm d, & \text{if } 1 \rightarrow 0 \text{ error occurred,} \end{cases} \tag{3. 10}$$

where $0 \leq d \leq m$.

By (3. 10) and the construction of CH_i's, we have the following results.

(i) If $2t + 2 \leq m_1 - d$, then

$$CH_i^{X''} < CH_i^{X}, \text{ for all } i = 1, \cdots, t + 1, \text{ if } 0 \rightarrow 1 \text{ error occurred,}$$
$$\text{or} \tag{3. 11}$$
$$CH_i^{X} < CH_i^{X''}, \text{ for all } i = 1, \cdots, t + 1, \text{ if } 1 \rightarrow 0 \text{ error occurred.}$$

(ii) If $m_1 - d \le 2t + 1$, then at least $\left\lfloor \frac{m_1 - d}{2} \right\rfloor$ i's in which we have

$$\begin{aligned} & CH_i^{X''} < CH_i^X, \text{ if } 0 \to 1 \text{ error occurred,} \\ \text{or} \quad & \\ & CH_i^X < CH_i^{X''}, \text{ if } 1 \to 0 \text{ error occurred.} \end{aligned} \tag{3. 12}$$

Recall that $0 \le d \le m$, then by (3. 8), (3. 4), (3. 11), and (3. 12), we have

$$\begin{aligned} t + 1 &\le \left\lfloor \frac{m_1 + m + m - d}{2} \right\rfloor \\ &= m + \left\lfloor \frac{m_1 - d}{2} \right\rfloor \\ &\le m + D_H((CH_1^X)' \cdots (CH_{t+1}^X)',\ CH_1^{X''} \cdots CH_{t+1}^{X''}) . \end{aligned}$$

Therefore, the algorithm signals "errors detected". //

3.6. On the Number of Checkbits and the Size of a Code

It has been mentioned in Chapter 2 that probably the most basic problem in coding theory is to find the most efficient code of given conditions, such as to find the largest code of given length and minimum distance (or some given error correcting/detecting properties), to find the minimum number of checkbits of a given length of the information and given error correcting/detecting properties in a systematic construction, or to find the maximum number of errors that can be corrected/detected for given lengths of the information and the check, etc.. In this section, the number of checkbits used in the proposed code and some previous methods will be examined. Also, bounds on the number

of checkbits and on the size of a code will be discussed.

It has been mentioned in Section 3.3 that the methods used in [59] and [75] are more efficient than the methods used in [9], [10], [19], and [62]. Thus, only the proposed method and methods used in [59] and [75] will be compared, Since all these three methods have similar structure, by adding $t + 1$ check symbols to some kind of t-error correcting code, let R_1, R_2, and R_3 denote the total number of bits in these $t + 1$ check symbols (i.e. $CH_1 \cdots CH_{t+1}$) in the proposed method, the method in [59], and the method in [75], respectively. Then,

$$R_1 = \sum_{i=1}^{t+1} \left\lceil \log_2 \left(\frac{(n+2) - [(2t+2) - 2(i-1)]}{2t+2} + 1 \right) \right\rceil ,$$

$$R_2 = \sum_{i=1}^{t+1} \left\lceil \log_2 \frac{n+1}{(2t+1) - 2(t-i+1)} \right\rceil ,$$

and $$R_3 = \sum_{i=1}^{t+1} \left\lceil \log_2 \frac{n+i}{t+1} \right\rceil ,$$

where n = the length of C, the linear t-error correcting systematic code with $D_H(C) = 2t + 1$. Recall that in the proposed method the construction started with C and its parity check bits, hence $n + 2$ instead of $n + 1$ appears in the formula R_1. Also, let n^* = the code length, then $n^* = n + 1 + R_1$ for the proposed method and $n^* = n + R_i$, $i = 2, 3$, for the methods in [59] and [75] respectively. It is evident that the proposed method has the shortest n^*. In Tables 3.2 and 3.3, which are adopted from [75] with some modification, the comparisons of n^* in all three different methods are presented for 2-EC/AUED codes and 3-EC/AUED codes. In the tables, n^* for the methods in [59] and [75] may be smaller than $n + R_i$, $i = 2, 3$, because of some modifications have been done (see [59] and [75]). Even with these modifications, the proposed method still shows smaller n^*. k denotes the number of information bits in Tables 3.2 and 3.3.

Table 3.2. 2-EC/AUED codes.

k	n	n*		
		Nikolos et. al.	Tao et. al.	Proposed
7	15	23	24	22
8	16	25	25	23
9	17	27	26	24
10	19	29	28	27
11	20	30	29	28
12	21	32	30	30
22	32	45	44	42
26	38	51	50	48
32	42	56	54	52
51	63	77	78	76
53	65	81	80	78
64	76	92	91	89
112	125	142	143	141
113	127	144	145	143

Table 3.3. 3-EC/AUED codes.

k	n	n*		
		Nikolos et. al.	Tao et. al.	Proposed
8	19	31	30	28
9	20	32	32	29
11	22	34	34	31
12	23	36	35	32
16	29	43	41	41
23	37	53	53	50
27	41	58	57	54
32	47	64	63	60
46	62	79	78	78
64	81	102	101	98
106	127	148	150	148
231	255	280	282	280

Now, let us consider how low the redundancy can be when constructing a t-EC/AUED code. A lower bound on the number of checkbits required for any systematic t-EC/AUED code has been given in [19], the only result so far. It is restated as the following theorem.

Theorem 3.7. For any systematic t-EC/AUED code, with k information bits and r checkbits, r must satisfy the following condition:

$$\log_2\left(2\left[1+\binom{k}{1}+\binom{k}{2}+\cdots+\binom{k}{t}\right]+\binom{k}{t+1}-\binom{2t+1}{t+1}\right) \le r.$$

This bound, when $t << k$, has asymptotically required $((t+1)\log_2 k)$ checkbits. The proposed method has saved quite a few bits when compared with any previous method, though, the asymptotical behavior of the proposed method and all other methods in [9], [10], [19], [59], and [75] require approximately $((2t+1)\log_2 k)$ checkbits (including checkbits in C). Thus, there is still a bit to go. By comparing the reasoning for getting the bound in Theorem 3.7 and the conditions in Theorem 3.1, we think that the bound can be pushed further. On the other hand, we think that to construct a t-EC/AUED code directly from information part may shorten the redundant check part. (Because we think that starting the construction from a t-error correcting systematic code, the proposed method almost reaches its extremity.) But, so far there is no significant result yet in either direction.

As to a bound of the maximum size of a t-EC/AUED code, in general including nonsystematic codes, no significant result has been found so far in the literature, except for the case of a SEC/AUED code. By applying the famous Sperner's theorem, Bose has derived a good bound for a SEC/AUED code, which will be stated in Theorem 4.17 of Chapter 4.

Chapter 4

Study of Bose-Rao Codes

4.1. Introduction

In Chapter 3 t-EC/AUED codes have been discussed and for practical purpose the discussion was concentrated on systematic construction, whereas an ingenious nonsystematic construction for a class of SEC/AUED codes given by Bose and Rao [20] plays an important role in error control. Therefore, a deeper study of Bose-Rao codes will be developed in this chapter.

First of all, let us review Bose-Rao codes construction. Let F_w^n be the set of $\binom{n}{w}$ binary vectors of length n and weight w, where $2 \le w \le n - 2$, and G be any Abelian group of order n. Suppose the elements of G be indexed as

$$g^{(0)} = 0, g^{(1)}, \cdots, g^{(n-1)}.$$

Define a mapping $T : F_w^n \rightarrow G$ as

$$T(X) = \sum_{i=0}^{n-1} x_i g^{(i)},$$

where $X = (x_0, x_1, \cdots, x_{n-1})$ and $x_i g^{(i)} = g^{(i)}$ if $x_i = 1$, or $g^{(0)}$, the summation is taking place in G. Then F_w^n is partitioned into n subsets,

$$V_i = \{X \in F_w^n \mid T(X) = g^{(i)}\}, i = 0, 1, \cdots, n - 1.$$

Bose and Rao have shown that each of these subsets is a SEC/AUED code. Further, they have observed that at least one of these subsets has cardinality greater than or equal to

$\frac{1}{n}\binom{n}{w}$. However, they have not given formulas for $|V_i|$'s and have not known what group G and which element $g^{(i)}$ of G make V_i the maximal size.

Here the technique which McEliece and Rodemich [57] used in analyzing Constantin-Rao codes [23] is applied to analyze Bose-Rao codes and the following results will be obtained (see Section 4.4).

(i) For $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ and a fixed $2 \le w \le n - 2$, take

$$G = (Z_{p_1} \oplus \cdots \oplus Z_{p_1}) \oplus \cdots \oplus (Z_{p_k} \oplus \cdots \oplus Z_{p_k})$$

and

$$g^{(i)} = \begin{cases} (1, 0, \cdots, 0), \text{ if } w \equiv 2 \pmod 4 \text{ and } (w, n) \equiv 2 \pmod 4, \\ 0, \qquad \text{otherwise,} \end{cases}$$

then $|V_i|$ is maximal over all possible Abelian groups and their elements.

The formula will be given in Theorem 4.14 of Section 4.4.

(ii) For $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ and $w = \lfloor \frac{n}{2} \rfloor$, take G as in (i) and

$$g^{(i)} = \begin{cases} (1, 0, \cdots, 0), \text{ if } \frac{n}{2} \equiv 2 \pmod 4, \\ 0, \qquad \text{otherwise,} \end{cases}$$

then $|V_i|$ is maximal over all possible values $2 \le w \le n - 2$, all possible Abelian groups and their elements. That is, the maximal size of Bose-Rao SEC/AUED code has been found.

The formula will be given in Theorem 4.16 of Section 4.4.

Further, the result (ii) will be compared with an upper bound of the size of a SEC/AUED code, which has been found by Bose (see Section 4.5). The background knowledge needed for deriving the above results will be discussed in Sections 4.2 and 4.3.

For readers' convenience a list of the notations used in this chapter is listed below.

Z_n : the (cyclic) additive group of integers modulo n.

o(g) : the order of an element g in a group G.

$G \oplus H$: the direct sum of Abelian groups G and H.

$\sum_{i=1}^{n} \oplus G_i$: the direct sum of Abelian groups $G_1, \cdots, G_n$.

$\sum_{1}^{n} \oplus G$: the direct sun of n copies of the Abelian group G.

F_w^n : the set of $\binom{n}{w}$ binary vectors of length n and weight w.

${}_GV_g^{(w)}$: Let G be an Abelian group of order n and their elements be indexed as $g^{(0)} = 0, g^{(1)}, \cdots, g^{(n-1)}$.

$${}_GV_g^{(w)} = \{X \in F_w^n \mid \sum_{i=0}^{n-1} x_i g^{(i)} = g\}, \text{ where } X = (x_0, x_1, \cdots, x_{n-1}).$$

If there is no ambiguity, this may be simply denoted as $V_g^{(w)}$, ${}_GV_g$, or V_g.

| A | : the cardinality of the set A.

a | b : the integer a divides the integer b.

a ∤ b : the integer a does not divide the integer b.

(a, b) : the g.c.d. of integers a and b.

E(p; n) : the highest power of p dividing n. e.g. If n = 24, then E(2; 24) = 3.

$\prod D$: D is a set of integers. $\prod D = \prod_{d \in D} d$.

e.g. D = {2, 3, 5}, then $\prod D = 2 \times 3 \times 5 = 30$.

A(n, d, w) : the maximum number of codewords in any binary code of length n, constant weight w, and Hamming distance d.

$N(X, Y)$: X and Y are two binary n-tuples. $N(X, Y)$ refers to the number of $1 \rightarrow 0$ crossovers from X to Y. For example, when $X = 1010$ and $Y = 0111$, it will have $N(X, Y) = 1$ and $N(Y, X) = 2$.

4.2. Some Knowledge from Fourier Analysis

It is well-known that any finite Abelian group G is isomorphic to a unique direct sum of cyclic groups of prime power order (references [25], [46], or [68].) Thus, any finite Abelian group G may be written as

$$G = \sum_{i=1}^{m} \oplus Z_{n_i}$$

with each n_i a prime power and hence any $g \in G$ may be written as $g = (g_1, \cdots, g_m)$ with $0 \le g_i < n_i$ for $i = 1, \cdots, m$. Let $g = (g_1, \cdots, g_m)$ and $h = (h_1, \cdots, h_m)$ be elements (not necessarily distinct) in G, define

$$< g, h > = \prod_{i=1}^{m} \zeta_i^{g_i h_i} ,$$

where ζ_i = a complex primitive n_i-th root of unity.

Lemma 4.1. The definition of $< g, h >$ satisfies the following properties.

(i) $< g, h > = < h, g >$.

(ii) $< g, h > < g, h' > = < g, h + h' >$.

(iii) $< g, jh > = < jg, h > = < g, h >^j$ for any integer j.

(iv) $$\sum_{g \in G} < g, h > = \begin{cases} 0, & \text{if } h \neq 0, \\ |G|, & \text{if } h = 0. \end{cases}$$

Proof. (i), (ii), and (iii) can be derived straightforward from the definition. If $h = 0$, then (iv) is obvious. If $h = (h_1, \cdots, h_m) \neq 0$, there is some $h_j \neq 0$, then

$$\sum_{g \in G} < g, h > = \sum_{g_j = 0}^{n_j - 1} \Big(\zeta_j^{g_j h_j} \sum_{(g_1, \cdots, g_j, \cdots, g_m) \in G} \Big(\prod_{i=1; i \neq j}^{m} \zeta_i^{g_i h_i} \Big) \Big)$$

$$= \Big(\sum_{g_j = 0}^{n_j - 1} \zeta_j^{h_j g_j} \Big) \Big(\sum_{(g_1, \cdots, g_m) \in G;\, g_j = 0} \Big(\prod_{i=1; i \neq j}^{m} \zeta_i^{g_i h_i} \Big) \Big)$$

$$= \frac{(\zeta_j^{h_j})^{n_j} - 1}{\zeta_j^{h_j} - 1} \Big(\sum_{(g_1, \cdots, g_m) \in G;\, g_j = 0} \Big(\prod_{i=1; i \neq j}^{m} \zeta_i^{g_i h_i} \Big) \Big)$$

$$= 0 . \; //$$

Now, let f be any function mapping G into the complex numbers. The Fourier transform $\hat{f}$ of f is defined as

$$\hat{f}(h) = \sum_{g \in G} \Big(< h, -g > f(g) \Big) .$$

Lemma 4.2. (Fourier inversion formula)

$$f(g) = \frac{1}{|G|} \sum_{h \in G} \Big(< h, g > \hat{f}(h) \Big) .$$

Proof. Using (ii) and (iv) of Lemma 4.1, we have

$$\frac{1}{|G|} \sum_{h \in G} \Big(< h, g > \hat{f}(h) \Big)$$

$$= \frac{1}{|G|} \sum_{h \in G} \left(< h,\ g > \sum_{g' \in G} \left(< h,\ -g' > f(g') \right) \right)$$

$$= \frac{1}{|G|} \sum_{h \in G} \sum_{g' \in G} \left(< h,\ g - g' > f(g') \right)$$

$$= \frac{1}{|G|} \sum_{g' \in G} \sum_{h \in G} \left(< h,\ g - g' > f(g') \right)$$

$$= \frac{1}{|G|} |G|\ f(g)$$

$$= f(g). \ //$$

Lemma 4.3. The mapping $g \rightarrow < -h,\ g >$ is a homomorphism of G onto the complex d-th roots of unity, where $d = o(h)$.

Proof. By (ii) and (iii) of Lemma 4.1, it is easy to see that the mapping is a homomorphism from G into the complex d-th roots of unity. Thus, it needs only to show that the mapping is an onto mapping.

Since $h = (h_1, \cdots, h_m) \in G$, where $G = \sum_{i=1}^{m} \oplus Z_{n_i}$ with each n_i a prime power, one can find a subset $\{i_1, \cdots, i_s\}$ of the set $\{1, \cdots, m\}$ such that $o(h_{i_j})$, $j = 1, \cdots, s$, are pairwise relatively prime, and $d = o(h) = \prod_{j=1}^{s} o(h_{i_j})$.

Let $g = (g_1, \cdots, g_m)$ with $g_i = 1$ if $i \in \{i_1, \cdots, i_s\}$, $g_i = 0$ otherwise. Then, $< -h,\ g >$ is a primitive d-th root of unity. Therefore, the mapping is onto. //

4.3. Some Knowledge from Combinatorics

Lemma 4.4.

(i) $\binom{n}{i} < \binom{n}{j}$ if $0 \le i < j \le \lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil \le j < i \le n$, for $2 \le n$.

(ii) $\binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{\lceil \frac{n}{2} \rceil}$ for $1 \le n$.

Proof. Both relations are well-known. Omitted. //

Lemma 4.5. If $0 < d_2 < d_1 \le n$, $0 \le w \le n$, and d_1, d_2 both divide n and w, then

$$\binom{n/d_1}{w/d_1} \le \binom{n/d_2}{w/d_2},$$

where equality holds only when $w = 0$ and n.

Proof. If $0 < w \le \lfloor \frac{n}{2} \rfloor$, then

$$\binom{n/d_1}{w/d_1} < \binom{n/d_2}{w/d_1} < \binom{n/d_2}{w/d_2}.$$

If $\lceil \frac{n}{2} \rceil \le w < n$, then $0 < n - w \le \lfloor \frac{n}{2} \rfloor$. Thus,

$$\binom{n/d_1}{w/d_1} = \binom{n/d_1}{(n-w)/d_1} < \binom{n/d_2}{(n-w)/d_2} = \binom{n/d_2}{w/d_2}.$$

It is obvious that when $w = 0$ and n the equality holds. //

Lemma 4.6.

$$\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n} \quad \text{for } 0 \le n.$$

Proof. We observe that the expression on the left-hand side is the constant term in $(1 + x)^n (1 + x^{-1})^n = x^{-n}(1 + x)^{2n}$. Thus, the equality holds. //

Lemma 4.7.

$$\binom{2n}{n-1} + \binom{n}{\lfloor \frac{n}{2} \rfloor} \leq \binom{2n}{n} \quad \text{for } 1 \leq n,$$

where equality holds only when $n = 1$ and 2.

Proof. A straightforward computation shows the inequality for $n = 3$ and the equality for $n = 1$ and 2. For $4 \leq n$, we have

$$\binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n} > \frac{1}{n+1}\binom{n}{\lfloor \frac{n}{2} \rfloor}^2 \qquad \text{(by Lemma 4.6)}$$

$$\geq \frac{1}{n+1}\binom{n}{2}\binom{n}{\lfloor \frac{n}{2} \rfloor} = \frac{n-1}{2+\frac{2}{n}}\binom{n}{\lfloor \frac{n}{2} \rfloor} > \binom{n}{\lfloor \frac{n}{2} \rfloor}. \; //$$

Lemma 4.8. If $0 < w < n$ and n, w both are even, then

$$(n-1)\binom{n/2}{w/2} \leq \binom{n}{w},$$

where equality holds only when $w = 2$ and $(n - 2)$.

Proof.

$$\binom{n}{w} = \binom{n/2}{w/2} \frac{(n-1)(n-3)\cdots(n-w+1)}{1 \cdot 3 \cdot \cdots \cdot (w-1)}$$

$$= \binom{n/2}{w/2} (n-1) \text{ (some number greater than or equal to 1)}$$

$$\geq \binom{n/2}{w/2} (n-1).$$

It is easy to see that the equality holds only when $w = 2$ and $(n - 2)$. //

Lemma 4.9.

$$\binom{n}{\frac{n}{2}-1} + 3\binom{n/2}{\frac{n}{4}-1} < \binom{n}{n/2} + \binom{n/2}{n/4}, \quad \text{for } 4 \le n \text{ and } 4 \mid n.$$

Proof.

$$\binom{n}{n/2} - \binom{n}{\frac{n}{2}-1} = \frac{2}{n+2}\binom{n}{n/2}. \tag{4.1}$$

$$3\binom{n/2}{\frac{n}{4}-1} - \binom{n/2}{n/4} = \frac{2n-4}{n+4}\binom{n/2}{n/4}. \tag{4.2}$$

It is easy to check that (4.2) < (4.1) for $n = 4$ and 8. For $12 \le n$, we have

$$\frac{2}{n+2}\binom{n}{n/2} - \frac{2n-4}{n+4}\binom{n/2}{n/4}$$

$$> \frac{2}{n+2}\binom{n/2}{n/4}^2 - \frac{2n-4}{n+4}\binom{n/2}{n/4} \qquad \text{(by Lemma 4.6)}$$

$$\ge \left(\frac{2}{n+2} \cdot \frac{\frac{n}{2}}{1} \cdot \frac{\frac{n}{2}-1}{2} \cdot \frac{\frac{n}{2}-2}{3} - \frac{2n-4}{n+4}\right)\binom{n/2}{n/4}$$

$$= \frac{n^4 - 2n^3 - 64n^2 + 32n + 192}{24n^2 + 144n - 196}\binom{n/2}{n/4}$$

$$\ge 0.$$

Therefore, the assertion is true. //

Lemma 4.10.

$$\binom{m}{\lfloor \frac{m}{2} \rfloor} = \binom{n}{\lfloor \frac{n}{2} \rfloor}, \quad \text{for } 0 < m < n.$$

Proof. This can be proved by induction on n. It is obvious that when $n = 2$ it is true. Assuming it is true for $(n - 1)$, then

$$\binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n-1}{\lfloor \frac{n}{2} \rfloor} + \binom{n-1}{\lfloor \frac{n}{2} \rfloor - 1} > \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}.$$

Therefore, the inequality holds. //

Lemma 4. 11. If n = odd and p = minimum prime dividing n, then

$$\binom{n}{(n-p)/2} + (n-1)\binom{n/p}{(n-p)/2p} \leq \binom{n}{\lfloor \frac{n}{2} \rfloor},$$

where equality holds only when n = 3.

Proof. If n is a prime then p = n. Thus,

$$\binom{p}{(p-p)/2} + (p-1)\binom{p/p}{(p-p)/2p} = p \leq \binom{p}{\lfloor \frac{p}{2} \rfloor},$$

where the equality holds only when p = 3.

If n = 9, it is easy to check that

$$\binom{n}{(n-3)/2} + (n-1)\binom{n/3}{(n-3)/6} < \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

If 15 ≤ n, then by Lemmas 4.4 and 4.10

$$\binom{n}{(n-p)/2} + (n-1)\binom{n/p}{(n-p)/2p} < \binom{n}{(n-3)/2} + (n-1)\binom{(n-1)/2}{\lfloor \frac{n-1}{4} \rfloor}.$$

Thus, it needs to be shown that the right-hand side of the above inequality is less than or equal to $\binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{(n-1)/2}$.

Equivalently, it will be shown that

$$\binom{(n-1)/2}{\lfloor \frac{n-1}{4} \rfloor} < \frac{1}{n-1}\left[\binom{n}{(n-1)/2} - \binom{n}{(n-3)/2}\right].$$

And,

$$\frac{1}{n-1}\left[\binom{n}{(n-1)/2} - \binom{n}{(n-3)/2}\right]$$

$$= \frac{8n}{n^3+3n^2-n-3}\binom{n-1}{(n-1)/2}$$

$$> \frac{8n}{n^3+3n^2-n-3}\binom{(n-1)/2}{\lfloor\frac{n-1}{4}\rfloor}^2 \qquad \text{(by Lemma 4.6)}$$

$$> \frac{8n}{n^3+3n^2-n-3}\binom{(n-1)/2}{\lfloor\frac{n-1}{4}\rfloor}\binom{(n-1)/2}{3}$$

$$= \frac{n-9+\frac{23}{n}-\frac{15}{n^2}}{6+\frac{18}{n}-\frac{6}{n^2}-\frac{3}{n^3}}\binom{(n-1)/2}{\lfloor\frac{n-1}{4}\rfloor}$$

$$> \binom{(n-1)/2}{\lfloor\frac{n-1}{4}\rfloor} . \ //$$

4.4. Analysis of Bose-Rao Codes

In this section the maximal size of Bose-Rao code, for both cases fixed w and over all possible w, will be discussed.

Let the prime factorization of n be $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$, i.e. $1 \le k$, $1 \le \alpha_i$ for $i = 1, \cdots, k$, and p_i's are distinct primes. If n is even, we may assume $p_1 = 2$. If G is an Abelian group of order n, as explained in Section 4.2, G may be written as

$$\left(\sum_{i=1}^{s_{p_1}} \oplus Z_{p_1^{\beta_{p_1 i}}}\right) \oplus \cdots \oplus \left(\sum_{i=1}^{s_{p_k}} \oplus Z_{p_k^{\beta_{p_k i}}}\right),$$

where $\sum_{i=1}^{s_{p_j}} \beta_{p_j i} = \alpha_j$, for $j = 1, \cdots, k$.

Also, E(p; n) will denote the highest power of p dividing n, e.g. if $p = p_i$, $E(p; n) = \alpha_i$. And, m will denote the number of summands of G.

Let f be a function mapping G into integers defined as $f(g) = |{}_G V_g^{(w)}|$ for some fixed w and $\hat{f}$ be the Fourier transform of f. $\hat{f}$ can be calculated as follows.

Lemma 4.12.

$$\hat{f}(h) = \begin{cases} (-1)^{(d+1)\frac{w}{d}} \binom{n/d}{w/d}, & \text{if } d \mid w, \\ 0, & \text{if } d \nmid w, \end{cases} \qquad \text{where } d = o(h).$$

Proof.

$$\hat{f}(h) = \sum_{g \in G} \left(< h, -g > f(g) \right)$$

$$= \sum_{g \in G} \left(< -h, g > f(g) \right) \qquad \text{(by (iii) of Lemma 4.1)}$$

$$= \sum_{X \in F_w^n} < -h, \sum_{i=0}^{n-1} x_i g^{(i)} > \qquad \text{(by the definition of f(g))}$$

$$= \sum_{X \in F_w^n} \left(\prod_{i=0}^{n-1} < -h, g^{(i)} >^{x_i} \right). \qquad \text{(by (ii) and (iii) of Lemma 4.1)}$$

By Lemma 4.3, we observe that the last quantity is the coefficient of x^w in

$$(1 + x)^{\frac{n}{d}} (1 + \zeta x)^{\frac{n}{d}} \cdots (1 + \zeta^{d-1} x)^{\frac{n}{d}} = (1 + (-1)^{d+1} x^d)^{\frac{n}{d}},$$

where ζ = a complex primitive d-th root of unity. Therefore, the formula holds. //

Using this formula and Fourier inversion formula, it is able to compute $|{}_GV_g^{(w)}|$ for any Abelian group G, any element g of G, and any value w. But we are more interested in optimal values of $|{}_GV_g^{(w)}|$'s for both fixed w and all possible G, g, and w.

Before developing the main theorems, we need one more formula.

Lemma 4.13. If n is even and G is an Abelian group of order n, then

$$\sum_{h\in G;\, o(h)=2} < h,\ g > = \begin{cases} 2^{s_{p_1}} - 1, & \text{if all } g_1 \text{ through } g_{s_{p_1}} \text{ in } g = (g_1, \cdots, g_m) \\ & \text{are even,} \\ -1, & \text{otherwise,} \end{cases}$$

for any $g \in G$.

Proof. If all g_1 through $g_{s_{p_1}}$ are even, then $< h,\ g > = 1$ for any $h \in G$ with $o(h) = 2$. And there are $(2^{s_{p_1}} - 1)$ elements of order two in G. Thus, the first part is proved.

Now, suppose not all g_1 through $g_{s_{p_1}}$ are even. Let $\{g_{i_1}, \cdots, g_{i_t}\}$ be the set of all those odd numbers in $\{g_1, \cdots, g_{s_{p_1}}\}$. For each $\{i_1, \cdots, i_t\} \supseteq A \neq \phi$, define

$$H_A = \{h \in G \mid h = (h_1, \cdots, h_m) \text{ where } h_i = 2^{(\beta_{p_1 i} - 1)} \text{ if } i \in A,\ h_i = 0 \text{ if}$$

$$i \in \{i_1, \cdots, i_t\} - A \text{ or } s_{p_1} < i \le m, \text{ and other } h_i\text{'s can be either } 0 \text{ or } 2^{(\beta_{p_1 i} - 1)}\}.$$

We have $|H_A| = 2^{(s_{p_1} - t)}$ and $< h,\ g > = (-1)^{|A|}$ for any $h \in H_A$.

Also, define

$$H_0 = \{h \in G \mid h = (h_1, \cdots, h_m) \text{ where } h_i = 0 \text{ if } i \in \{i_1, \cdots, i_t\} \text{ or}$$

$$s_{p_1} < i \le m, \text{ and other } h_i\text{'s can be either } 0 \text{ or } 2^{(\beta_{p_1 i} - 1)}\} - \{(0, \cdots, 0)\}.$$

We have $|H_\phi| = 2^{(s_{p_1} - t)} - 1$ and $< h, g > = 1$ for any $h \in H_\phi$.

We observe that

$$\{ H_A \mid \{i_1, \cdots, i_t\} \supseteq A \neq \phi \} \cup \{ H_\phi \}$$

forms a partition of the set $\{ h \in G \mid o(h) = 2 \}$. Thus,

$$\sum_{h \in G;\, o(h)=2} < h, g >$$

$$= \sum_{\{i_1, \cdots, i_t\} \supseteq A \neq \phi} \Big(\sum_{h \in H_A} < h, g > \Big) + \sum_{h \in H_\phi} < h, g >$$

$$= \sum_{\{i_1, \cdots, i_t\} \supseteq A \neq \phi} \Big((-1)^{|A|} 2^{(s_{p_1} - t)} \Big) + 2^{(s_{p_1} - t)} - 1$$

$$= \Big(\sum_{i=0}^{t} \binom{t}{i} (-1)^i \Big) 2^{(s_{p_1} - t)} - 1$$

$$= -1.$$

Therefore, the second part is proved, too. //

Now, we have enough tools to tackle our main problems.

Theorem 4.14. For a fixed $2 \leq w \leq n - 2$, $|_G V_g|$ is maximized by

$$G = \Big(\sum_{1}^{\alpha_1} \oplus Z_{p_1} \Big) \oplus \cdots \oplus \Big(\sum_{1}^{\alpha_k} \oplus Z_{p_k} \Big)$$

and

$$g = \begin{cases} (1, 0, \cdots, 0), & \text{if } w \equiv 2 \pmod 4 \text{ and } (w, n) \equiv 2 \pmod 4, \\ 0, & \text{otherwise.} \end{cases}$$

Let $|V|_{max} = \max_{G,\, g} |{}_G V_g|$, the maximum is taking over all possible Abelian group G of order n and all $g \in G$. Then $|V|_{max}$ is formulated as follows.

(i) If $(w, n) = 1$, then $|V|_{max} = \frac{1}{n}\binom{n}{w}$.

(ii) If $(w, n) \neq 1$ and $w \not\equiv 2 \pmod 4$ or $(w, n) = q_1^{\beta_1} \cdots q_s^{\beta_s} \not\equiv 2 \pmod 4$, where $\{p_1, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$, then

$$|V|_{max} = \frac{1}{n}\left[\binom{n}{w} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D}\left(p^{E(p;n)} - 1\right)\right)\binom{n/\Pi D}{w/\Pi D}\right]. \tag{4. 3}$$

(iii) If $(w, n) \neq 1$, $w \equiv 2 \pmod 4$, and $(w, n) = 2\, q_1^{\beta_1} \cdots q_s^{\beta_s} \equiv 2 \pmod 4$, where $\{p_2, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$, then

$$|V|_{max} =$$

$$\frac{1}{n}\left[\binom{n}{w} + \binom{n/2}{w/2} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D}\left(p^{E(p;n)} - 1\right)\right)\left(\binom{n/\Pi D}{w/\Pi D} + \binom{n/2\Pi D}{w/2\Pi D}\right)\right]. \tag{4. 4}$$

(Note: In (iii), $\{q_1, \cdots, q_s\}$ can be empty. If that is the case then the third term on the right-hand side of (4. 4) is dropped out. That is $|V|_{max} = \frac{1}{n}\left[\binom{n}{w} + \binom{n/2}{w/2}\right]$.)

Proof.

Case (i) $(w, n) = 1$.

By Lemmas 4.2 and 4.12, it is easy to see that in this case $|{}_G V_g| = \frac{1}{n}\binom{n}{w}$ for any Abelian group G and any $g \in G$.

Case (ii) $(w, n) \neq 1$ and $w \not\equiv 2 \pmod 4$ or $(w, n) = q_1^{\beta_1} \cdots q_s^{\beta_s} \not\equiv 2 \pmod 4$,

where $\{p_1, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$.

Take the group G as stated in the theorem and $g = 0$, then by Lemmas 4.2 and 4.12, we have

$$|_G V_g| = \frac{1}{n}\left[\binom{n}{w} + \sum_{0 \neq h \in G;\, o(h) \mid w} \hat{f}(h)\right]$$

$$= \text{the right-hand side of (4. 3).}$$

The last equality holds because for any $0 \neq h \in G$ with $o(h) \mid w$ iff there exists $\{q_1, \cdots, q_s\} \supseteq D \neq \phi$ such that $o(h) = \Pi D$. Besides, there are $\prod_{p \in D} (p^{E(p; n)} - 1)$ elements in G having order ΠD.

Next, it needs to show that the right-hand side of (4. 3) is greater than or equal to $|_H V_h|$ for all possible Abelian group H of order n and all $h \in H$.

Let $f(h) = |_H V_h|$. For each $\{q_1, \cdots, q_s\} \supseteq D \neq \phi$, define

$$H_D = \{h \in H \mid o(h) \mid w \text{ and } o(h) = \prod_{p \in D} p^{\gamma_p} \text{ with } 1 \leq \gamma_p \text{ for all } p \in D\}.$$

By Lemmas 4.12 and 4.5, we have

$$|\hat{f}(h)| = \binom{n/o(h)}{w/o(h)} \leq \binom{n/\Pi D}{w/\Pi D} \text{ for any } h \in H_D. \tag{4. 5}$$

Also, note that

$$|H_D| \leq \prod_{p \in D} (p^{E(p; n)} - 1). \tag{4. 6}$$

Besides, $\{H_D \mid \{q_1, \cdots, q_s\} \supseteq D \neq \phi\}$ forms a partition of $\{h \in H \mid o(h) \mid w \text{ and } o(h) \neq 1\}$.

Thus, for any $h \in H$, we have

$$f(h) = |f(h)|$$

$$\leq \frac{1}{n}\left[\binom{n}{w} + |\hat{f}(h^{(1)})| + \cdots + |\hat{f}(h^{(n-1)})|\right]$$

(by the definition of f and the triangle inequality)

$$= \frac{1}{n}\left[\binom{n}{w} + \sum_{0 \neq h \in H;\, o(h) | w} |\hat{f}(h)|\right]$$

$$= \frac{1}{n}\left[\binom{n}{w} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\sum_{h \in H_D} |\hat{f}(h)|\right)\right]$$

$$\leq \frac{1}{n}\left[\binom{n}{w} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D} (p^{E(p;n)} - 1)\right)\binom{n/\Pi D}{w/\Pi D}\right].$$

(by (4. 5) and (4. 6))

Case (iii) $(w, n) \neq 1$, $w \equiv 2 \pmod 4$, and $(w, n) = 2\, q_1^{\beta_1} \cdots q_s^{\beta_s} \equiv 2 \pmod 4$,

where $\{p_2, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$.

Let G be an arbitrary Abelian group of order n and be written as

$$\left(\sum_{i=1}^{s_{p_1}} \oplus Z_{p_1^{\beta_{p_1 i}}}\right) \oplus \cdots \oplus \left(\sum_{i=1}^{s_{p_k}} \oplus Z_{p_k^{\beta_{p_k i}}}\right),$$

where $\sum_{i=1}^{s_{p_j}} \beta_{p_j i} = \alpha_j$, for $j = 1, \cdots, k$, and $p_1 = 2$.

And, $f(g)$ denotes $|{}_G V_g|$.

By the properties of w and n, Lemmas 4.2 and 4.12, we have

f(g)

$$= \frac{1}{n}\left[\binom{n}{w} - \sum_{h \in G;\, o(h)=2} \langle h,\ g \rangle \binom{n/2}{w/2} + \sum_{h \in G;\, 1,2 \neq o(h) \mid w} (-1)^{o(h)+1} \langle h,\ g \rangle \binom{n/o(h)}{w/o(h)}\right]$$

$$= \frac{1}{n}\left[\binom{n}{w} - \sum_{h \in G;\, o(h)=2} \langle h,\ g \rangle \binom{n/2}{w/2} + \sum_{h \in G;\, 1 \neq o(h)=\text{odd};\, o(h) \mid w} \langle h,\ g \rangle \binom{n/o(h)}{w/o(h)} - \sum_{h \in G;\, 2 \neq o(h)=\text{even};\, o(h) \mid w} \langle h,\ g \rangle \binom{n/o(h)}{w/o(h)}\right]$$

$$= \frac{1}{n}\left[\binom{n}{w} - \sum_{h \in G;\, o(h)=2} \langle h,\ g \rangle \binom{n/2}{w/2} + \sum_{h \in G;\, 1 \neq o(h)=\text{odd};\, o(h) \mid w} \langle h,\ g \rangle \binom{n/o(h)}{w/o(h)} - \sum_{h \in G;\, 1 \neq o(h)=\text{odd};\, o(h) \mid w} \left(\sum_{h' \in G;\, o(h')=2} \langle (h'+h),\ g \rangle \right) \binom{n/2o(h)}{w/2o(h)}\right]$$

$$= \frac{1}{n}\Big[\binom{n}{w} - \sum_{h\in G;\, o(h)=2} < h,\ g > \binom{n/2}{w/2}$$

$$+ \sum_{h\in G;\, 1\neq o(h)=\text{odd};\, o(h)\mid w} < h,\ g > \binom{n/o(h)}{w/o(h)}$$

$$- \sum_{h\in G;\, 1\neq o(h)=\text{odd};\, o(h)\mid w} < h,\ g > \Big(\sum_{h'\in G;\, o(h')=2} < h',\ g >\Big)\binom{n/2o(h)}{w/2o(h)}\Big]$$

(by (ii) of Lemma 4.1)

$$= \begin{cases} \frac{1}{n}\Big[\binom{n}{w} - (2^{s_{p_1}} - 1)\binom{n/2}{w/2} \\ \quad + \sum_{h\in G;\, 1\neq o(h)=\text{odd};\, o(h)\mid w} < h,\ g > \Big(\binom{n/o(h)}{w/o(h)} - (2^{s_{p_1}} - 1)\binom{n/2o(h)}{w/2o(h)}\Big)\Big], \\ \quad \text{if all } g_1 \text{ through } g_{s_{p_1}} \text{ in } g = (g_1, \cdots, g_m) \text{ are even,} \\ \frac{1}{n}\Big[\binom{n}{w} + \binom{n/2}{w/2} \\ \quad + \sum_{h\in G;\, 1\neq o(h)=\text{odd};\, o(h)\mid w} < h,\ g > \Big(\binom{n/o(h)}{w/o(h)} + \binom{n/2o(h)}{w/2o(h)}\Big)\Big], \\ \quad \text{otherwise.} \end{cases}$$

(by Lemma 4.13)

(4. 7)

Therefore, if $G = \Big(\sum_1^{\alpha_1} \oplus Z_{p_1}\Big) \oplus \cdots \oplus \Big(\sum_1^{\alpha_k} \oplus Z_{p_k}\Big)$ and $g = (1, 0, \cdots, 0)$ we have $f(g)$ = the right-hand side of (4.4).

Next, it needs to show that the right-hand side of (4. 4) is greater than or equal to

$|{}_H V_h|$ for all possible Abelian group H of order n and all $h \in H$. For the rest of the proof $f(h)$ will denote $|{}_H V_h|$.

Similar to case (ii), for each $\{q_1, \cdots, q_s\} \supseteq D \neq \phi$, define

$$H_D = \{h \in H \mid o(h) | w \text{ and } o(h) = \prod_{p \in D} p^{\gamma_p} \text{ with } 1 \leq \gamma_p \text{ for all } p \in D\}.$$

We have

$$|H_D| \leq \prod_{p \in D} (p^{E(p; n)} - 1). \tag{4. 8}$$

Also, by Lemma 4.5,

$$\binom{n/o(h)}{w/o(h)} \leq \binom{n/\Pi D}{w/\Pi D} \text{ and } \binom{n/2o(h)}{w/2o(h)} \leq \binom{n/2\Pi D}{w/2\Pi D}$$

$$\text{for any } h \in H_D. \tag{4. 9}$$

Besides, by Lemma 4.8, it can be seen that if $o(h) = \text{odd}$ and $o(h) \mid w$, we have

$$(2^{s_{p_1}} - 1)\binom{n/2o(h)}{w/2o(h)} \leq (\frac{n}{o(h)} - 1)\binom{n/2o(h)}{w/2o(h)} < \binom{n/o(h)}{w/o(h)}. \tag{4. 10}$$

Thus, by (4. 7), (4. 8), (4. 9), and (4. 10), for any $h \in H$, we have

$$f(h)$$

$$\leq \frac{1}{n}\left[\binom{n}{w} + \binom{n/2}{w/2} + \sum_{h \in H;\, 1 \neq o(h) = \text{odd};\, o(h) | w} \left(\binom{n/o(h)}{w/o(h)} + \binom{n/2o(h)}{w/2o(h)}\right)\right],$$

(by (4. 7) and (4. 10))

$$= \frac{1}{n}\left[\binom{n}{w} + \binom{n/2}{w/2} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \sum_{h \in H_D} \left(\binom{n/o(h)}{w/o(h)} + \binom{n/2o(h)}{w/2o(h)}\right)\right]$$

$\leq$ the right-hand side of (4. 4). (by (4. 8) and (4. 9))

//

Remarks:

(i) Note that the group and the element of the group in the theorem is not the only possibility which makes the Bose-Rao code maximal. For instance, when $n = 2^r$, $3 \leq r$, and $(w, n) = 2$, any Abelian group G and any $g = (g_1, \cdots, g_m) \neq 0$ with at least one $g_i \neq$ even make Bose-Rao code maximal. Since our interest was merely on finding the maximal size of Bose-Rao code, the theorem has not exhausted all such groups and elements of the group that make the Bose-Rao code maximal. A little further study of the proof of the theorem and the properties of all possible Abelian groups of order n should easily bring out all such groups and elements.

(ii) When $w = 2$ and $n - 2$ both cases (ii) and (iii) in the theorem get the same value $|V|_{max} = \frac{n}{2}$.

(iii) Since $(w, n) = (n - w, n)$ it can be shown that

$$\max |{}_G V_g^{(w)}| = \max |{}_G V_g^{(n-w)}|.$$

(iv) Since Bose-Rao codes form binary codes of length n of constant weight w and Hamming distance four, the value in the theorem becomes a lower bound for $A(n, 4, w)$. Some values in the theorem give better lower bounds than those given by Graham and Sloane [33].

e.g. $2710 \leq A(18, 4, 9)$, $6330 \leq A(20, 4, 8)$,

$30789 \leq A(24, 4, 8)$, $112952 \leq A(24, 4, 12)$.

These were found independently by Klϕve [44]. //

Next, the maximal size of Bose-Rao code over all possible value w will be

investigated. Here, a lemma in number theory will be needed.

Lemma 4.15. If $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ and $\{p_1, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$, then

$$\sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D} \left(p^{E(p;n)} - 1 \right) \right) \leq n - 1 .$$

Proof. The inequality will be proved by induction on k. If $k = 1$, it is obvious that the inequality holds. Assuming the inequality holds for $k - 1$, then, let $n = n_1 p_k^{\alpha_k}$, we have

$$\sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D} \left(p^{E(p;n)} - 1 \right) \right)$$

$$\leq \sum_{\{p_1, \cdots, p_k\} \supseteq D \neq \phi} \left(\prod_{p \in D} \left(p^{E(p;n)} - 1 \right) \right)$$

$$= \sum_{\{p_1, \cdots, p_{k-1}\} \supseteq D \neq \phi} \left(\prod_{p \in D} \left(p^{E(p;n)} - 1 \right) \right)$$

$$+ \left[1 + \sum_{\{p_1, \cdots, p_{k-1}\} \supseteq D \neq \phi} \left(\prod_{p \in D} \left(p^{E(p;n)} - 1 \right) \right) \right] \left(p_k^{\alpha_k} - 1 \right)$$

$$\leq (n_1 - 1) + n_1 (p_k^{\alpha_k} - 1) \qquad \text{(by induction hypotheses)}$$

$$= n - 1 .$$

Thus, the inequality holds. //

Theorem 4.16. For all $2 \le w \le n-2$, G, and $g \in G$, $|{}_G V_g^{(w)}|$ is maximized by

$$w = \lfloor \tfrac{n}{2} \rfloor, \quad G = \Big(\sum_1^{\alpha_1} \oplus Z_{p_1} \Big) \oplus \cdots \oplus \Big(\sum_1^{\alpha_k} \oplus Z_{p_k} \Big),$$

and

$$g = \begin{cases} (1, 0, \cdots, 0), & \text{if } \frac{n}{2} \equiv 2 \pmod 4, \\ 0, & \text{otherwise.} \end{cases}$$

And, their values are

(i) $\frac{1}{n} \binom{n}{\lfloor \frac{n}{2} \rfloor}$, if $n = \text{odd}$,

(ii) $$\frac{1}{n} \Big[\binom{n}{n/2} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \Big(\prod_{p \in D} (p^{E(p;n)} - 1) \Big) \binom{n/\Pi D}{n/2\Pi D} \Big],$$

if $n = \text{even}$ and $\frac{n}{2} \not\equiv 2 \pmod 4$;

where $\{p_1, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$ and $(n, \frac{n}{2}) = q_1^{\beta_1} \cdots q_s^{\beta_s}$, (4. 11)

(iii) $$\frac{1}{n} \Big[\binom{n}{n/2} + \binom{n/2}{n/4} + \sum_{\{p_2, \cdots, p_k\} \supseteq D \neq \phi} \Big(\prod_{p \in D} (p^{E(p;n)} - 1) \Big) \Big(\binom{n/\Pi D}{n/2\Pi D} + \binom{n/2\Pi D}{n/2\Pi D} \Big) \Big],$$

if $n = \text{even}$ and $\frac{n}{2} \equiv 2 \pmod 4$. (4. 12)

Proof. By Theorem 4.14, it is only needed to compare $|V^{(w)}|_{max}$ for all different values $2 \le w \le n-2$. Once the value of w is determined, the choices of G, g, and the formula are direct results of Theorem 4.14. It will be discussed in three cases.

Case (i) $n = \text{odd}$.

If (w, n) = 1, then

$$\max_{G,\,g} |{}_G V_g^{(w)}| = \frac{1}{n}\binom{n}{w} \le \frac{1}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}. \qquad \text{(by (i) of Theorem 4.14)}$$

If $(w, n) \neq 1$, let $p_0 = \min\{p_1, \cdots, p_k\}$, then

$$\max_{G,\,g} |{}_G V_g^{(w)}|$$

$$\le \frac{1}{n}\left[\binom{n}{w} + \sum_{\{q_1, \cdots, q_s\} \supseteq D \neq \phi} \left(\prod_{p \in D}\left(p^{E(p;n)} - 1\right)\right)\binom{n/p_0}{w/p_0}\right]$$

where $\{p_1, \cdots, p_k\} \supseteq \{q_1, \cdots, q_s\}$

(by (ii) of Theorem 4.14 and Lemma 4.5)

$$\le \frac{1}{n}\left[\binom{n}{w} + (n-1)\binom{n/p_0}{w/p_0}\right] \qquad \text{(by Lemma 4.15)}$$

$$\le \frac{1}{n}\left[\binom{n}{(n-p_0)/2} + (n-1)\binom{n/p_0}{(n-p_0)/2p_0}\right] \qquad \text{(by Lemma 4.4)}$$

since $(w, n) \neq 1$ implies $1 < w \le (n - p_0)/2$ or $(n + p_0)/2 \le w < n$

$$\le \frac{1}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}. \qquad \text{(by Lemma 4.11)}$$

Further, note that $(\lfloor \frac{n}{2} \rfloor, n) = 1$.

Therefore, the theorem holds for this case.

Case (ii) n = even and $\frac{n}{2} \not\equiv 2 \pmod 4$.

First of all, note that in this case $p_1 = 2$ and $\alpha_1 = 1$ or $3 \le \alpha_1$.

Thus,

$$(n,\ \frac{n}{2}) = \begin{cases} p_2^{\alpha_2} \cdots p_k^{\alpha_k}, & \text{if } \alpha_1 = 1, \\ p_1^{\alpha_1 - 1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}, & \text{if } 3 \le \alpha_1. \end{cases} \tag{4. 13}$$

Now, let us compare (4. 11) with all $\max_{G, g} |_G V_g^{(w)}|$.

If $(w, n) = 1$ or $w \not\equiv 2 \pmod 4$, then by (i) and (ii) of Theorem 4.14 and (4. 13)

$$\max_{G, g} |_G V_g^{(w)}| \le (4. 11)$$

is straightforward.

If $w \equiv 2 \pmod 4$, then

$$(w, n) = 2\, d_1^{\gamma_1} \cdots d_t^{\gamma_t} \quad \text{with} \quad \{p_2, \cdots, p_k\} \supseteq \{d_1, \cdots, d_t\}.$$

Also, note that $1 < w \le \frac{n}{2} - 1$ or $\frac{n}{2} + 1 \le w < n$.

Thus, by Lemma 4.4 and Lemma 4.7, we have

$$\binom{n}{w} + \binom{n/2}{w/2} \le \binom{n}{\frac{n}{2} - 1} + \binom{n/2}{\lfloor \frac{n}{4} \rfloor} \le \binom{n}{n/2} \tag{4. 14}$$

and

$$\binom{n/\Pi D}{w/\Pi D} + \binom{n/2\Pi D}{w/2\Pi D} \le \binom{n/\Pi D}{\frac{n}{2\Pi D} - 1} + \binom{n/2\Pi D}{\lfloor n/4\Pi D \rfloor} \le \binom{n/\Pi D}{n/2\Pi D} \tag{4. 15}$$

where $\{p_2, \cdots, p_k\} \supseteq \{d_1, \cdots, d_t\} \supseteq D$.

Then, by (iii) of Theorem 4.14, (4. 14), and (4. 15)

$$\max_{G,\,g} |{}_G V_g^{(w)}|$$

$$= \frac{1}{n}\left[\binom{n}{w} + \binom{n/2}{w/4} + \sum_{\{d_1, \cdots, d_t\} \supseteq D \neq \phi} \left(\prod_{p \in D}\left(p^{E(p;n)} - 1\right)\right)\left(\binom{n/\Pi D}{w/\Pi D} + \binom{n/2\Pi D}{w/2\Pi D}\right)\right]$$

$$\leq \frac{1}{n}\left[\binom{n}{n/2} + \sum_{\{p_2, \cdots, p_k\} \supseteq D \neq \phi} \left(\prod_{p \in D}\left(p^{E(p;n)} - 1\right)\right)\binom{n/\Pi D}{n/2\Pi D}\right]$$

$\leq$ (4. 11).

Therefore, the theorem holds for this case.

Case (iii) n = even and $\frac{n}{2} \equiv 2 \pmod 4$.

Note that in this case, $p_1 = 2$, $\alpha_1 = 2$, and $(n, \frac{n}{2}) = 2\, p_2^{\alpha_2} \cdots p_k^{\alpha_k} \equiv 2 \pmod 4$.

So, by (iii) of Theorrem 4.14, we have

$$\max_{G,\,g} |{}_G V_g^{(\frac{n}{2})}| = (4.\ 12).$$

Next, it is needed to compare (4. 12) with all $\max_{G,\,g} |{}_G V_g^{(w)}|$.

By (i) of Theorem 4.14, the case (w, n) = 1 is trivial. It is needed only to discuss the case $(w, n) \neq 1$. If $w \equiv 2 \pmod 4$ then $(w, n) \equiv 2 \pmod 4$.

Thus, by (iii) of Theorem 4.14, the result is obvious.

If $w \not\equiv 2 \pmod 4$, then

$$(w, n) = \begin{cases} d_1^{\gamma_1} \cdots d_t^{\gamma_t} , & \text{if } w = \text{odd}, \\ 4\, d_1^{\gamma_1} \cdots d_t^{\gamma_t} , & \text{if } w = \text{even}, \end{cases}$$

where $\{p_2, \cdots, p_k\} \supseteq \{d_1, \cdots, d_t\}$.

If w = odd then by (ii) of Theorem 4.14 and Lemma 4.4 the result is clear, too. If w = even, then by (ii) of Theorem 4.14 and Lemma 4.9,

$$\max_{G,\, g} |{}_G V_g^{(w)}|$$

$$= \frac{1}{n}\left[\binom{n}{w} + \sum_{\{p_1, d_1, \cdots, d_t\} \supseteq D \neq \phi} \left(\prod_{p \in D} (p^{E(p;\, n)} - 1)\right) \binom{n/\Pi D}{w/\Pi D}\right]$$

$$= \frac{1}{n}\left[\binom{n}{w} + \sum_{\{d_1, \cdots, d_t\} \supseteq D \neq \phi} \left(\left(\prod_{p \in D} (p^{E(p;n)} - 1)\right)\left(\binom{n/\Pi D}{w/\Pi D} + 3\binom{n/2\Pi D}{w/2\Pi D}\right)\right) + 3\binom{n/2}{w/2}\right]$$

$\leq$ (4. 12). (by Lemma 4.4 and Lemma 4.9)

Therefore, the proof of the theorem is completed. //

Remarks:

(i) Similar to Theorem 4.14, the third term in (iii) of Theorem 4.16 may be dropped out. But this happens only when n = 4.

(ii) Also, the group and the element of the group in Theorem 4.16 is not the only possibility which makes the Bose-Rao code maximal.

4.5. On the Size of a SEC/AUED Code

In this section the superiority of Bose-Rao codes will be examined, In order to explore this, it needs to know an upper bound on the size of a SEC/AUED code. Bose has extended the famous Sperner's Lemma [72] (also can be found in [34] and [50]) to obtain the following theorem.

Theorem 4.17. [Bose]* The number of codewords in a SEC/AUED code with length n is no more than $\frac{2}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}$.

(* Note: The result has not been published yet.)

Proof. Let C be a SEC/AUED code which gives maximum number of codewords. The total number of bits in C will be $n|C|$. In C, either the total number of 0's or the total number of 1's will be at least $\frac{n}{2}|C|$. Without loss of generality, it may assume that the total number of 1's in C will be at least $\frac{n}{2}|C|$. For $X \in C$, let S_X represent the set of all vectors obtained by a single $1 \rightarrow 0$ crossover from X. Since C is capable of correcting single errors, for any $X, Y \in C$ with $X \neq Y$ implies $S_X \cap S_Y = \phi$.

Let $S = \bigcup_{X \in C} S_X$. Then, $\frac{n}{2}|C| \leq |S|$.

Furthermore, for any X_1, $X_2 \in S$, with $X_1 \neq X_2$ it will have $1 \leq N(X_1, X_2)$ and $1 \leq N(X_2, X_1)$. (i.e. The elements in S are unordered.)
Therefore, by Sperner's Lemma it will have $|S| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}$.
That is, $|C| \leq \frac{2}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}$. //

From Theorem 4.16, we knew that the maximal Bose-Rao code has size between $\frac{1}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}$ and $\frac{2}{n}\binom{n}{\lfloor \frac{n}{2} \rfloor}$. Thus, these group theoretic SEC/AUED codes are close to optimal. Besides, if one examines the proof of Theorem 4.17 and the statement of Sperner's Lemma more closely, one may feel that the upper bound found by Bose can be improved. According to Sperner's Lemma, $|S|$ in the proof of Theorem 4.17 reaches to $\binom{n}{\lfloor \frac{n}{2} \rfloor}$ only when S contains all weight $\lfloor \frac{n}{2} \rfloor$ (or all weight $\lceil \frac{n}{2} \rceil$) elements, which is obviously not the case from the structure of S. This convinces us that more than likely Bose-Rao codes are almost optimal.

Chapter 5

t-Error Correcting and d-Unidirectional Error Detecting Codes

5.1. Introduction

It has been mentioned in Chapter 3 that error correcting/detecting codes that are effective against both symmetric and unidirectional errors are useful in providing protection against transient, intermittent, and permanent faults. In Chapter 3 the discussion was concentrated on t-EC/AUED codes, whereas for some practical performances we may not need to detect all errors but only up to some limited d unidirectional errors. Here $t + 2 \leq d$ is considered, since any t symmetric error correcting code can be easily and efficiently converted, by adding one even (or odd) parity checkbit, to a t symmetric error correcting and $t + 1$ symmetric error detecting code. Of course, we expect a better code (i.e. higher rate or equivalently less redundant bits being used) for t-error correcting/d-unidirectional error detecting(t-EC/d-UED) code than for t-EC/AUED code. A little discussion and a construction for single error correcting/d-unidirectional error detecting(SEC/d-UED) code have been brought up in [11]. Besides this, no other investigation has been done in this area so far.

The basic requirements for a code being t-EC/d-UED are reviewed in Section 5.2. A nonsystematic code construction for SEC/d-UEC code is developed in Section 5.3. And, systematic code constructions for SEC/d-UED code and t-EC/d-UED ($2 \leq t$) code are constructed in Section 5.4. Then, a decoding algorithm of the code constructed in Section 5.4 is developed in Section 5.5. Last, a bound on the number of redundant bits for the systematic code and the number of redundant bits used for the proposed code in Section

5.4 are discussed in Section 5.6. Also, an upper bound on the size of a SEC/d-UED code is developed in Section 5.6.

5.2. The Basic Theorem of t-EC/d-UED Code

The following fundamental theorem describes the necessary and sufficient conditions for a t-EC/d-UEC code [11].

Theorem 5.1. A code is t-EC/d-UED iff it satisfies the following condition:

for all $X, Y \in C$ with $X \neq Y$ implies

either $t + d + 1 \leq D_H(X, Y)$

or $t + 1 \leq N(X, Y)$ and $t + 1 \leq N(Y, X)$.

By the same reason as SEC/AUED codes playing more important role in all t-EC/AUED codes, the SEC/d-UEC codes will be expected to be more popular and more important among all t-EC/d-UED codes. So, when $t = 1$, Theorem 5.1 is restated as the following corollary.

Corollary 5.2. A code C is SEC/d-UED iff it satisfies the following condition:

for all $X, Y \in C$ with $X \neq Y$ implies

either $d + 2 \leq D_H(X, Y)$

or $2 \leq N(X, Y)$ and $2 \leq N(Y, X)$.

5.3. Nonsystematic Code Construction for SEC/d-UED Code

There is no significant nonsystematic t-EC/d-UED ($2 \le t$) code so far. In this section only nonsystematic SEC/AUED codes will be constructed. The code will be constructed is an extension to the Bose-Rao code described in Chapter 4. Given n and w, $2 \le w \le n - 2$, let $V^{(w)}$ be the maximal Bose-Rao code obtained in Theorem 4.14. For a fixed d, $3 \le d \le \lceil \frac{n}{2} \rceil - 4$, define

$$V_i = \bigcup_{j=0}^{\lfloor \frac{n-i-2}{d+2} \rfloor} V^{(i + j(d+2))} ,$$

for $i = 2, \cdots, d + 3$.

Theorem 5.3. Each V_i, $i = 2, \cdots, d + 3$, is SEC/d-UED.

Proof. Let $X, Y \in V_i$, then $X \in V^{(i + j_1(d+2))}$ for some j_1 and $Y \in V^{(i + j_2(d+2))}$ for some j_2. If $j_1 = j_2$, then by the property of Bose-Rao code structure $2 \le N(X, Y)$ and $2 \le N(Y, X)$. If $j_1 \ne j_2$, then $d + 2 \le D_H(X, Y)$. Therefore, by Corollary 5.2 V_i is SEC/d-UEC. //

A few words have to be said about the value of d. If some V_i contains merely a single $V^{(w)}$ for some w, then V_i is not only SEC/d-UED but SEC/AUED. Besides, if this is the case then the construction makes no sense. (Because a better code has not reached.) Therefore, d was restricted to $3 \le d \le \lceil \frac{n}{2} \rceil - 4$ so that each V_i contains more than one $V^{(w)}$. At least the V_i which contains $V^{(\lfloor \frac{n}{2} \rfloor)}$ (or $V^{(\lceil \frac{n}{2} \rceil)}$) as a subset is strictly larger than $V^{(\lfloor \frac{n}{2} \rfloor)}$ (recall that $|V^{(\lfloor \frac{n}{2} \rfloor)}| = |V^{(\lceil \frac{n}{2} \rceil)}|$). Then, by Theorem 4.16, the construction does get better codes than maximal Bose-Rao codes.

It has not been found yet which of these V_i, $i = 2, \cdots, d + 3$, has the maximal

size. We conjecture that the V_i which has $V^{(\lfloor \frac{n}{2} \rfloor)}$ or $V^{(\lceil \frac{n}{2} \rceil)}$ as a subset has the maximal size. The following example explains this situation.

Example 5.1. Let $n = 32$. The sizes of $V^{(w)}$, $2 \le w \le 16$, are shown below. Recall that $|V^{(w)}| = |V^{(n-w)}|$. (See remark (iii) of Theorem 4.14.)

w	2	3	4	5	6	7	8	9	10
$\|V^{(w)}\|$	16	155	1240	6293	28336	105183	330460	876525	2016144

w	11	12	13	14	15	16
$\|V^{(w)}\|$	4032015	7063784	10855425	14732720	17678835	18796230

Now, the sizes of V_i's are compared for five different values of d.

For $d = 3$, $|V_4| = |V_3|$, $|V_5| = |V_2|$, $V_6 \supset V^{(16)}$, and

i	2	3	6
$\|V_i\|$	26870255	26746525	26916932

For $d = 4$, $|V_5| = |V_3|$, $|V_6| = |V_2|$, $V_4 \supset V^{(16)}$, and

i	2	3	4	7
$\|V_i\|$	22155316	22593823	22830998	21921216

For $d = 5$, $|V_6| = |V_5|$, $|V_7| = |V_4|$, $|V_8| = |V_3|$, $V_2 \supset V^{(16)}$, and

i	2	3	4	5
$\lvert V_i \rvert$	20549282	20025594	18871158	17953838

.

For d = 6, $\lvert V_5 \rvert = \lvert V_3 \rvert$, $\lvert V_6 \rvert = \lvert V_2 \rvert$, $\lvert V_9 \rvert = \lvert V_7 \rvert$, $V_8 \supset V^{(16)}$, and

i	2	3	4	7	8
$\lvert V_i \rvert$	16777216	14893888	14130048	18660543	19457150

.

For d = 7, $\lvert V_3 \rvert = \lvert V_2 \rvert$, $\lvert V_8 \rvert = \lvert V_6 \rvert$, $\lvert V_9 \rvert = \lvert V_5 \rvert$, $\lvert V_{10} \rvert = \lvert V_4 \rvert$, $V_7 \supset V^{(16)}$, and

i	2	4	5	6	7
$\lvert V_i \rvert$	11095970	12872809	15615538	18037631	19006596

.

Although the maximal size of V_i, $i = 2, \cdots, d + 3$, is unknown, there is one thing for sure. By Theorem 4.14, we have

$$\frac{1}{n} \sum_{2 \le \lfloor \frac{n}{2} \rfloor \pm j(d+2) \le n-2} \binom{n}{\lfloor \frac{n}{2} \rfloor \pm j(d+2)} \le \max_{2 \le i \le d+3} \lvert V_i \rvert . \qquad (5.1)$$

5.4. Systematic Code Construction for t-EC/d-UED (1 ≤ t) Code

In this section, three different methods of constructing systematic t-EC/d-UED (1≤ t) code will be developed. To construct a systematic t-EC/d-UED code, some bits are appended to a t symmetric error correcting and t + 1 symmetric error detecting code (i.e. a systematic parity check code with Hamming distance 2t + 2.) Let C be a

systematic parity check code with $D_H(C) = 2t + 2$, then the t-EC/d-UED code constructed from C, denoted as C', will have this form:

$$C' = \{ X\, CH_X \mid X \in C \text{ and } CH_X \text{ is the appended symbol}\},$$

where CH_X may be simply denoted as CH if there is no ambiguity.

How powerful C' can be, or equivalently how large d can be will depend on the choice of the symbol CH and the number of bits in CH. The description of CH will be discussed in three cases.

Method A: If there exists a set of s-bit symbols , called S, such that $D_H(S) = m$ and $\frac{2t + m + 2}{2} \leq |S|$, then use s bits for CH to construct a t-EC/(t + m + 1)-UED code.

Method B: Use s bits, here $t + 3 \leq s$, for CH to construct a t-EC/(2s - t - 1)-UED code.

Method C: If there exists a t-EC/AUED code S with length s such that $s < |S|$ then use s bits for CH to construct a t-EC/(2 |S| - t - 1)-UED code.

Method A

Theorem 5.4. Let S be a set of s-bit symbols such that $D_H(S) = m$ and $\frac{2t + m + 2}{2} \leq |S|$. Also, let the elements of S be indexed as

$$S = \{Z_0, Z_1, \cdots, Z_{|S|-1}\}.$$

Define a function f from $\{0, 1, \cdots, |S| - 1\}$ to S as $f(i) = Z_i$.

If CH_X is assigned as

$$CH_X = f(\frac{W_H(X)}{2} \bmod |S|) \text{ for all } X \in C,$$

then C' is t-EC/(t + m + 1)-UED.

Proof. Let $X' = X\,CH_X$ and $Y' = Y\,CH_Y$ be any two codewords in C', and $q = W_H(X) - W_H(Y)$. (It may assume $0 \le q$.) By Lemma 3.4, we have the following results.

(i) If $2t + m + 2 \le q$, then $2t + m + 2 \le D_H(X, Y)$, and hence $2t + m + 2 \le D_H(X', Y')$.

(ii) If $q = 0$ then $t + 1 \le N(X, Y)$ and $t + 1 \le N(Y, X)$, and hence $t + 1 \le N(X', Y')$ and $t + 1 \le N(Y', X')$.

(iii) If $2 \le q \le 2t + m + 1$, then $2t + 2 \le D_H(X, Y)$.

By these results and Theorem 5.1, it is necessary only to consider the situation $2 \le q \le 2t + m + 1$. Since $2 \le q \le 2t + m + 1$ and $\frac{2t + m + 2}{2} \le |S|$, $CH_X \ne CH_Y$. Thus, $m \le D_H(CH_X, CH_Y)$. Combine this with (iii), $2t + m + 2 \le D_H(X', Y')$ when $2 \le q \le 2t + m + 1$. Therefore, C' is t-EC/(t + m + 1)-UED. //

Let us consider the special case $m = 1$ in Theorem 5.4. Let $s = \lceil \log_2(t + 2) \rceil$ then the set, S, of all s-bit symbols always satisfies $D_H(S) = 1$ and $\frac{2t + 1 + 2}{2} \le t + 2 \le |S| = 2^s$. Therefore, we have the following corollary.

Corollary 5.5. Let $s = \lceil \log_2(t + 2) \rceil$. Define a function f from $\{0, 1, \cdots, 2^s - 1\}$ to s-bit symbols as

f(i) = the binary representation of i in s bits.

If CH_X is assigned as

$$CH_X = f(\frac{W_H(X)}{2} \bmod 2^s) \text{ for all } X \in C,$$

then C' is t-EC/(t+2)-UED.

By Corollary 5.5, a SEC/3-UED code can be constructed from the extended Hamming code by merely adding two bits to each codeword.

Corollary 5.6. There exists a systematic SEC/3-UED code with code length two longer than the extended Hamming code.

Now, consider the set $S = \{001, 010, 100\}$. S satisfies $D_H(S) = 2$ and $\frac{2 \cdot 1 + 2 + 2}{2} \leq |S|$. Thus, by Theorem 5.4, we have the following corollary.

Corollary 5.7. There exists a SEC/4-UED code using three bits for CH.

Method B

Theorem 5.8. Let $t + 3 \leq s$. Define

$$Z_0 = 0 \cdots 01 \cdots 1, \text{ there are } \lceil \tfrac{s}{2} \rceil \text{ 1's and } \lfloor \tfrac{s}{2} \rfloor \text{ 0's}$$

and

$$S = \{ Z_i \mid i = 0, 1, \cdots, s - 1 \text{ and } Z_{j+1} = \text{one bit (circularly) left shift of } Z_j, \; 0 \leq j \leq s - 2\}.$$

(e.g. If $t = 1$ and $s = 4$, then $Z_0 = 0011$, $Z_1 = 0110$, $Z_2 = 1100$, and $Z_3 = 1001$.)

Also, define a function f from $\{0, 1, \cdots, s - 1\}$ to S as $f(i) = Z_i$.

If CH_X is assigned as

$$CH_X = f(\frac{W_H(X)}{2} \bmod s) \text{ for all } X \in C,$$

then C' is t-EC/(2s - t - 1)-UED.

Proof. First of all, notice that f has the following properties:

$$\text{(1) } N(f(i), f(j)) = m, \text{ if } |i - j| = m \text{ or } s - m, \text{ where } m = 0, 1, \cdots, \min(t, \lfloor \tfrac{s}{2} \rfloor), \tag{5.2}$$

(2) $N(f(i), f(j)) \geq t + 1$, otherwise.

Let X', Y', and q be the same notations as in the proof of Theorem 5.4. By Lemma 3.4, we have the following results.

(i) If $2s \leq q$, then $2s \leq D_H(X, Y)$, and hence $2s \leq D_H(X', Y')$.

(ii) If $q = 0$, then $t + 1 \leq N(X, Y)$ and $t + 1 \leq N(Y, X)$, and hence $t + 1 \leq N(X', Y')$ and $t + 1 \leq N(Y', X')$.

(iii) If $2 \leq q \leq 2t$, then

$$\frac{2t + 2 + q}{2} \leq N(X, Y) \quad \text{and} \quad \frac{2t + 2 - q}{2} \leq N(Y, X).$$

(iv) If $2t + 2 \leq q \leq 2s - 2$, then $q \leq N(X, Y)$ and $0 \leq N(Y, X)$.

By these results and Theorem 5.1, further discussion is needed for the situation $2 \leq q \leq 2s - 2$. Let

$$i = \frac{W_H(X)}{2} \bmod s \quad \text{and} \quad j = \frac{W_H(Y)}{2} \bmod s.$$

Then, when $2 \leq q \leq 2s - 2$, we have

$$i - j = \frac{q}{2} \quad \text{or} \quad -(s - \frac{q}{2}). \tag{5.3}$$

The discussion for the situation $2 \leq q \leq 2s - 2$ will be divided into two cases, according to the value of s.

Case (i) $t + 3 \leq s \leq 2t + 1$.

In this case, $\min(t, \lfloor \frac{s}{2} \rfloor) = \lfloor \frac{s}{2} \rfloor$. Now, the discussion will be done on two subintervals $2 \leq q \leq s$ and $s + 1 \leq q \leq 2s - 2$ separately instead of the whole interval $2 \leq q \leq 2s - 2$.

(a) $2 \le q \le s$.

By (5. 2) and (5. 3), we have

$$\frac{q}{2} \le N(CH_X, CH_Y) \quad \text{and} \quad \frac{q}{2} \le N(CH_Y, CH_X). \tag{5. 4}$$

Combine (5. 4) with (iii), we have

$$t + 1 \le N(X', Y') \quad \text{and} \quad t + 1 \le N(Y', X').$$

(b) $s + 1 \le q \le 2s - 2$.

The condition of q implies

$$s - \frac{s-1}{2} \le \frac{q}{2} = s - (s - \frac{q}{2}) \le s - 1 \text{ and } 1 \le s - \frac{q}{2} \le \frac{s-1}{2}.$$

Thus, by (5. 2) and (5. 3), we have

$$s - \frac{q}{2} \le N(CH_X, CH_Y) \quad \text{and} \quad s - \frac{q}{2} \le N(CH_Y, CH_X). \tag{5. 5}$$

Then, combine (5. 5) with (iii) and (iv), we have

$$D_H(X', Y') = N(X, Y) + N(Y, X) + N(CH_X, CH_Y) + N(CH_Y, CH_X)$$
$$\ge 2s.$$

Therefore, when $t + 3 \le s \le 2t + 1$, the condition in Theorem 5.1 is satisfied for the situation $2 \le q \le 2s - 2$.

Case (ii) $2t + 2 \le s$.

In this case, $\min(t, \lfloor \frac{s}{2} \rfloor) = t$. Similar to Case (i), the discussion will be on three subintervals $2 \le q \le 2t$, $2t + 2 \le q \le 2s - 2t - 2$, and $2s - 2t \le q \le 2s - 2$ separately. (Recall that q is an even number.)

(a) $2 \le q \le 2t$.

The condition of q implies $1 \le \frac{q}{2} \le t$.

Thus, by (5. 2) and (5. 3), we have

$$\frac{q}{2} \le N(CH_X,\ CH_Y) \quad \text{and} \quad \frac{q}{2} \le N(CH_Y,\ CH_X). \tag{5. 6}$$

Combine (5. 6) with (iii), we have

$$t + 1 \le N(X',\ Y') \quad \text{and} \quad t + 1 \le N(Y',\ X').$$

(b) $2t + 2 \le q \le 2s - 2t - 2.$

The condition of q implies $t + 1 \le \frac{q}{2} \le s - (t + 1).$

Thus, by (5. 2) and (5. 3), we have

$$t + 1 \le N(CH_X,\ CH_Y) \quad \text{and} \quad t + 1 \le N(CH_Y,\ CH_X).$$

Hence, $t + 1 \le N(X',\ Y')$ and $t + 1 \le N(Y',\ X').$

(c) $2s - 2t \le q \le 2s - 2.$

The condition of q implies

$$t + 2 \le s - t \le \frac{q}{2} = s - (s - \frac{q}{2}) \le s - 1 \text{ and } 1 \le s - \frac{q}{2} \le t.$$

Thus, by (5. 2) and (5. 3), we have

$$s - \frac{q}{2} \le N(CH_X,\ CH_Y) \quad \text{and} \quad s - \frac{q}{2} \le N(CH_Y,\ CH_X). \tag{5. 7}$$

Combine (5. 7) with (iv), we have $2s \le D_H\ (X',\ Y').$

Therefore, when $2t + 2 \le s,$ the condition in Theorem 5.1 is satisfied for the situation $2 \le q \le 2s - 2,$ too.

Thus, the proof of the theorem is completed. //

Method C

Theorem 5.9. Let S be a t-EC/AUED code of length s. Also, let the elements of S be indexed as

$$S = \{Z_0, Z_1, \cdots, Z_{|S|-1}\}.$$

Define a function f from $\{0, 1, \cdots, |S| - 1\}$ to S as $f(i) = Z_i$.

If CH_X is assigned as

$$CH_X = f(\frac{W_H(X)}{2} \bmod |S|) \quad \text{for all } X \in C,$$

then C' is t-EC/(2 | S | - t - 1)-UED.

(Note: Naturally, one would choose S as large as possible and s as small as possible if one would use this method to construct a t-EC/d-UED code. Also, one would like to have $s \leq |S|$, or Method B would be used.)

Proof. Let X', Y', and q be the same notations as in the proof of Theorem 5.4.

Since S is t-EC/AUED, we have

$$t + 1 \leq N(Z_i, Z_j) \quad \text{for any } i \neq j. \tag{5. 8}$$

By Lemma 3.4, we have the following results.

(i) If $2|S| \leq q$, then $2|S| \leq q \leq N(X, Y)$, and hence $2|S| \leq D_H(X', Y')$.

(ii) If $q = 0$, then $t + 1 \leq N(X, Y)$ and $t + 1 \leq N(Y, X)$, and hence
$t + 1 \leq N(X', Y')$ and $t + 1 \leq N(Y', X')$.

(iii) If $2 \leq q \leq 2|S| - 2$, then $CH_X \neq CH_Y$ and hence by (5. 8)
$t + 1 \leq N(X', Y')$ and $t + 1 \leq N(Y', X')$.

Therefore, by Theorem 5.1, C' is t-EC/(2| S | - t - 1)-UED. //

Remarks (on Methods A, B, and C):

(1) Notice that when $t = 1$ and $8 \leq s$, by Theorem 4.16 and the table of A(n, 4, w) in [51], there always exists a SEC/AUED code S with length s such that $s \leq |S|$. Thus, when $8 \leq s$, Method C is the best method among all three methods to construct a SEC/d-UEC code.

In Table 5.1, the number of unidirectional errors, d, can be detected by the

SEC/d-UED codes, using the proposed methods, is shown for $2 \le s \le 15$, where s = the number of bits in CH. In the table, Method A is used for s = 2, 3, Method B is used for $4 \le s \le 7$, and Method C is used for $8 \le s \le 15$. Also, when $8 \le s \le 15$, the size of S(the SEC/AUED code of length s used for CH) is from the table of A(n, 4, w) in [51].

Table 5.1. The values of d using s bits for CH in SEC/d-UED codes.

s	2	3	4	5	6	7	8	9	10	11	12	13	14	15
d	3	4	6	8	10	12	26	34	70	130	262	284	430	868

In general, it can be seen that Method B is better than Method A when $2t + 2 \le s$. When $t + 3 \le s \le 2t + 1$ Method A may be better than Method B. But, for $2t + 2 \le s$, Method C is better than both Methods A and B. In Tables 5.2 -5.6, the values of d using s bits in CH for t = 2, 3, 4, 5, 6 are shown. Whenever the Method C is used in these tables, the values of | S |'s can be refered to the tables of A(n, 6, w), A(n, 8, w), and A(n, 10,w) in [51]. Also, some of the values of | S |'s in Method A are from the tables of A(n, d) in the same book [51].

In Table 5.2(t = 2), Methods A, B, and C are used for s = 2 and 4, $5 \le s \le 11$, and $12 \le s \le 15$, respectively. Notice that when s = 11, both Methods B and C give the same d = 19.

Table 5.2. The values of d using s bits for CH in 2-EC/d-UED codes.

s	2	4	5	6	7	8	9	10	11	12	13	14	15
d	4	5	7	9	11	13	15	17	19	41	49	81	137

In Table 5.3(t = 3), Methods A, B, and C are used for s = 3 and 4, $6 \leq s \leq 15$, and s = 16 and 17, respectively. Notice that when s = 15, both Methods B and C give the same d = 26.

Table 5.3. The values of d using s bits for CH in 3-EC/d-UED codes.

s	3	4	6	7	8	9	10	11	12	13	14	15	16	17
d	5	6	8	10	12	14	16	18	20	22	24	26	56	64

In Table 5.4(t = 4), Methods A, B, and C are used for s = 3, 4, 6, $8 \leq s \leq 19$, and s = 20, respectively. Note that for d = 8, if Method B is used it needs s = 7.

Table 5.4. The values of d using s bits for CH in 4-EC/d-UED codes.

s	3	4	6	8	9	.	.	.	.	.	.	.	.	18	19	20
d	6	7	8	11	13	.	.	.	.	.	.	.	.	31	33	71

In Table 5.5(t = 5), Methods A and B are used for s = 3, 4, 6, 7 and s = 9, 10, respectively. Note that for d = 10, if Method B is used it needs s = 8.

Table 5.5. The values of d using s bits for CH in 5-EC/d-UED codes.

s	3	4	6	7	9	10
d	7	8	9	10	12	14

In Table 5.6(t = 6), Methods A and B are used for s = 3, 5, 7, 8, 9 and s = 11, 12, respectively. Note that for d = 11, if Method B is used it needs s = 9 and for d = 12, if Method B is used it needs s = 10.

Table 5.6. The values of d using s bits for CH in 6-EC/d-UED codes.

s	3	5	7	8	9	11	12
d	8	9	10	11	12	15	17

(2) The number of bits used for CH is independent of n, the code length of C, contrast to the number of bits used for CH_1, CH_2, $\cdots$, and CH_{t+1} in t-EC/AUED code constructed in Chapter 3, which is dependent of n. Also, it has been mentioned in Section 5.1 that it is expected a better code for t-EC/d-UED than for t-EC/AUED. The comparison of R_1 in Section 3.6 and Tables 5.1-5.6 tells us these methods indeed do the job. For instance, if n = 64, the SEC/AUED code in Chapter 3 needs 10 bits forCH_1 andCH_2 . But, if we need to detect only up to 34 errors the proposed SEC/d-UED code uses fewer bits for CH (see Table 5.1.) Similarly, if n = 128, the 2-EC/AUED code in Chapter 3 needs 15 bits for CH_1, CH_2, and CH_3. But, if we need to detect only up to 81 errors the proposed 2-EC/d-UED code uses fewer bits for CH (see Table 5.2.)

(3) The number of bits used for CH may be reduced for some special code C. For instance, if C is the extended Hamming (8, 4) code, then the codewords in C have only three different weights, 0, 4, and 8. Thus, if we use CH = 0 for weight zero codeword and weight eight codeword, and CH = 1 for all weight four codewords, then C' is indeed SEC/3-UED. But, in general, reducing the number of bits in CH may not always be feasible. Here, we would like to show that to extend the extended Hamming (16, 11) code to SEC/3-UEDD, SEC/4-UED, and SED/5-UED code we need at least 2, 3, and 4 bits, respectively, in CH.

Let C be the extended Hamming (16, 11) code with the parity check matrix

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

Then,

$$\begin{aligned} X_1 &= 1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1, \\ X_2 &= 1\,1\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1, \\ X_3 &= 1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1, \\ X_4 &= 1\,1\,0\,1\,0\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,1, \quad \text{and} \\ X_5 &= 1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,1 \end{aligned}$$

are codewords in C with

$$N(X_{i+1},\ X_i) = 3,\quad N(X_i,\ X_{i+1}) = 1,\ \text{for}\ i = 1, 2, 3, 4,$$

and

$$N(X_{i+2},\ X_i) = 4,\quad N(X_i,\ X_{i+2}) = 0,\ \text{for}\ i = 1, 2, 3.$$

Case (i) need at least two bits for CH to extend C to SEC/3-UED.

In order to extend C to a SEC/3-UED code, X_1, X_2, and X_3 have to have distinct CH. Therefore, it needs at least 2 bits for CH.

Case (ii) need at least three bits for CH to extend C to SEC/4-UED.

Suppose C can be extended to a SEC/4-UED code by using two bits for CH.

Now, consider CH_{X_3}. If $CH_{X_3} = 00$ then CH_{X_4} has to be 11.

But $CH_{X_4} = 11$ forces $CH_{X_2} = 00$ which is impossible, since CH_{X_2} can not be same as CH_{X_3}. Also, $CH_{X_3} = 11$ forces $CH_{X_1} = 00$ and $CH_{X_2} = 00$ which is impossible, since CH_{X_1} and CH_{X_2} have to be distinct.

If $CH_{X_3} = 01$, then $CH_{X_1} = 10$. Thus, no proper symbol can be assigned to CH_{X_2}. By the same reason $CH_{X_3} = 10$ is also impossible. Therefore, in order to extend C to be SEC/4-UED, it needs at least three bits for CH.

Case (iii) need at least four bits for CH to extend C to SEC/5-UED.

Suppose C can be extended to a SEC/5-UED code by using three bits for CH.

Same as case (ii), let us consider all possibilities of CH_{X_3}.

If $CH_{X_3} = 000$, then both CH_{X_4} and CH_{X_5} have to be 111 which is impossible. Similarly, $CH_{X_3} = 111$ forces CH_{X_1} and CH_{X_2} both have to be 000 which is impossible. If $CH_{X_3} = 001$, then $CH_{X_5} = 110$. Thus, no proper symbol can be assigned to CH_{X_4}. Similarly, CH_{X_3} can not be 010 or 100. If $CH_{X_3} = 011$, then $CH_{X_1} = 100$. So, no proper symbol can be assigned to CH_{X_2}. Similarly, CH_{X_3} can not be 101 or 110.

Therefore, it needs at least four bits to extend C to a SEC/5-UED code. //

5.5. Decoding Algorithm

Since all three Methods A, B, and C in Section 5.4 use the same principle, except using different set, S, for the check symbol CH. In this section only the decoding algorithm for Method C is developed. As a matter of fact, the algorithm developed here can be applied to all three methods, except the proof of the validity needs slight modification.

Let $X^* = X\,CH_X$ be an error free transmitted codeword in the proposed t-EC/d-UED code and $(X^*)' = X'(CH_X)'$ be the received word with some errors in X^*.

Decoding Algorithm

(1) Compute the syndrome of X' as usual in code C. Let m be the multiplicity

of errors corresponding to the syndrome.

(2) If $t < m$ then signal "errors detected" and stop.

(3) Decode X' using a decoding algorithm in code C to get X" and compute $CH_{X"}$ for X".

(4) If $m + D_H((CH_X)', CH_{X"}) \leq t$, then

output X" $CH_{X"}$ and stop

else

signal "errors detected" and stop.

End(of Decoding Algorithm).

Theorem 5.10. The Decoding Algorithm described above is valid.

Proof. To prove the validity of the algorithm, it is necessary to prove that

(i) if t or fewer errors have occurred in the received word, then the algorithm outputs the correct codeword,

and

(ii) if more than t but no more than $d(= 2|S| - t - 1)$ unidirectional errors have occurred in the received word, the algorithm should signal "errors detected".

Let m_1 and m_2 be the numbers of errors have occurred in X and CH_X, respectively.

Case (i) t or fewer errors.

By the structure of C, $m_1 \leq t$ implies $m = m_1$ and $X" = X$ in steps (1) and (3).

Then, in step (4), $m + D_H((CH_X)', CH_{X"}) = m_1 + m_2 \leq t$.

Therefore, the algorithm outputs the correct codeword $X"CH_{X"} = XCH_X$.

Case (ii) more than t but no more than d unidirectional errors.

If $t < m$, then step (2) does the job. So, it is needed only to consider $m \leq t$.

What needs to be shown is $t < m + D_H((CH_X)', CH_{X"})$ for this case.

Two subcases will be discussed here.

Subcase (1) $m_1 \leq t$.

Same argument as in Case (i), $m + D_H((CH_X)', CH_{X''}) = m_1 + m_2$ in step (4). Because $t < m_1 + m_2$, therefore, the algorithm signals "errors detected".

Subcase (2) $t < m_1$.

By the structure of S(being t-EC/AUED) and the characteristic of unidirectional errors from CH_X to $(CH_X)'$, it is easy to see that if $CH_{X''} \neq CH_X$ then $t < D_H((CH_X)', CH_{X''})$. It will be shown that under the condition $t < m_1 \leq d$ and $m \leq t$, $CH_{X''} \neq CH_X$ always holds.

If $m = 0$, then $X'' = X' = X + Y$ with $W_H(Y) = m_1$. That is, $D_H(X'', X) = m_1$. Since $t < m_1 \leq d = 2^{|S|} - t - 1$, according to the definition of CH, $CH_{X''} \neq CH_X$.

If $0 < m \leq t$, then $X'' = X' + A$ with $W_H(A) = m$. On the other hand, $X' = X + B$ with $W_H(B) = m_1$. Thus, $X'' = X + A + B$, and hence

$$m_1 - m \leq D_H(X'', X) = W_H(A + B) \leq m_1 + m. \tag{5.9}$$

Since $t < m_1 \leq d = 2^{|S|} - t - 1$ and $0 < m \leq t$, we have

$$m_1 + m \leq 2^{|S|} - 1, \tag{5.10}$$

and

$$1 \leq m_1 - m. \tag{5.11}$$

In (5.11), $m_1 - m = 1$ happens only when $m = t$ and $m_1 = t + 1$. But since C is a t-error correcting and (t+1)-error detecting code, this never occurs. Thus, (5.11) can be rewritten as

$$2 \leq m_1 - m. \tag{5.12}$$

Now, combining (5.9), (5.10), with (5.12), we have

$$2 \leq D_H(X'', X) \leq 2^{|S|} - 1.$$

Thus, by the definition of CH, $CH_{X''} \neq CH_X$.

Therefore, the algorithm signals "errors detected". //

5.6. On the Number of Checkbits and the Size of a Code

As any other type of systematic code, the number of checkbits always is a big issue. In this section a lower bound on the number of checkbits in a t-EC/d-UED code will be developed. And, when $t = 1$, the number of checkbits used in the code constructed in Section 5.4 will be examined and compared with the code constructed in [11].

Furthermore, an upper bound on the size of a SEC/d-UED code is developed here, too. Then, the size of the code constructed in Section 5.3 is compared with this bound.

First of all, a lower bound on the number of checkbits is described in the following theorem.

Theorem 5.11. For any systematic t-EC/d-UED code ($1 \leq t$) with k information bits and r checkbits, r must satisfy the following condition:

$$\log_2 \left[\sum_{i=0}^{t} \binom{k}{i} + \binom{k}{t+1} - \binom{k-d+t}{t+1} \right] \leq r .$$

Proof. Consider the following sets of k-tuple information symbols.

$B_0 = \{ I = (i_0, \cdots, i_{k-1}) \mid W_H(I) \leq t \}$, and

$B_j = \{ I = (i_0, \cdots, i_{k-1}) \mid i_s = 1, \text{ for } 0 \leq s \leq j-1, \text{ and } W_H(I) = t + j \}$,

for $1 \leq j \leq d - t$.

Now, define $B = \bigcup_{i=0}^{d-t} B_i$.

By Theorem 5.1, it is easy to see that all information symbols in B need distinct check symbols. And

$$|B| = \sum_{i=0}^{t} \binom{k}{i} + \sum_{i=1}^{d-t} \binom{k-i}{t} = \sum_{i=0}^{t} \binom{k}{i} + \binom{k}{t+1} - \binom{k-d+t}{t+1} .$$

Therefore, the condition holds. //

For $t = 1$, the lower bound in the theorem is $\log_2\left(\frac{d(2k - d + 1) + 2}{2}\right)$; which is approximately equal to $\log_2(k) + \log_2(d)$.

The number of checkbits used in the SEC/d-UED code described in Section 5.4 is approximately equal to $\log_2(k) + (1 + s)$, where s = the length of CH in the proposed code. Thus, in Method C (in Section 5.4), if $s << |S|$ the code is close to optimal.

On the other hand, let us look at the code construction in [11]. There two check symbols are appended to information symbol. The first check symbol requires approximately $2\log_2(k)$ bits and the second check symbol requires approximately $\log_2(d)$ bits. Thus, this construction uses $\log_2(k)$ bits more than the bound developed in Theorem 5.11.

However, for moderate information length the proposed code is better than the code in [11].

Next, similar to Theorem 4.17, Bose has extended Borden's bound [8] on d-unidirectional error detecting code to obtain an upper bound on the size of a SEC/d-UED code.

Theorem 5.12. [Bose]* The number of codewords in a SEC/d-UED code with length n is no more than $\frac{2}{n}\left(\sum_{w \equiv \lfloor n/2 \rfloor \bmod d} \binom{n}{w}\right)$.

(* Note: The result has not been published yet.)

Proof. Let C be a SEC/d-UED code which gives the maximum number of codewords. The total number of bits in C will be $n|C|$. In C, either the total number of 0's or the total number of 1's will be at least $\frac{n}{2}|C|$. Without loss of generality, assume that the total number of 1's in C will be at leas $\frac{n}{2}|C|$. For $X \in C$, let S_X represent the set of all vectors obtained by a single $1 \rightarrow 0$ crossover from X. Since C is capable of correcting single errors, for any $X, Y \in C$ with $X \neq Y$ implies $S_X \cap S_Y = \phi$.

Let $S = \bigcup_{X \in C} S_X$. Then, $\frac{n}{2}|C| \leq |S|$.

Furthermore, for any $X_1, X_2 \in S$, with $X_1 \neq X_2$ it will have either $1 \leq N(X_1, X_2)$ and $1 \leq N(X_2, X_1)$ or $d \leq D_H(X_1, X_2)$. Thus, S forms a (d-1)-unidirectional error detecting code.

Therefore, by Borden's bound, it will have $|S| \leq \sum_{w \equiv \lfloor n/2 \rfloor \bmod d} \binom{n}{w}$.

That is, $|C| \leq \frac{2}{n}\left(\sum_{w \equiv \lfloor n/2 \rfloor \bmod d} \binom{n}{w} \right)$. //

From (5. 1), we knew that the code constructed in Section 5.3 has size at least close to half of the bound in this theorem.

Chapter 6

Conclusion

6.1. Summary and Future Research Efforts

In Chapter 2, we investigated the optimality of the codes constructed in [17] which are capable of detecting up to $5 \cdot 2^{r-4} + r - 4$ unidirectional errors by using r checkbits independent of the number of information bits. We found that the maximal number of unidirectional errors which can be detected by a systematic code using r checkbits is $2^r - 2^{\lceil \frac{r}{2} \rceil} - 2^{\lfloor \frac{r}{2} \rfloor} + 1$ and $2^r - 2^{\lceil \frac{r}{2} \rceil} - 3 \cdot 2^{\lfloor \frac{r}{2} \rfloor - 1} + 2$ for $k = 2^r$ and $k = 2^r + 1$, respectively. By these two values and some informal checking, we feel that when k is increasing the number of unidirectional errors which can be detected by a systematic code using r checkbits is decreasing. The codes constructed in [40] shows this tendency, too. Thus, we wonder again whether the codes constructed in [17] are optimal if k is greater than some number. Therefore, we would like to investigate whether there exists some number M such that when $M \leq k$ the maximal number of unidirectional errors which can be detected by a systematic code using r checkbits becomes a constant, and if this constant is equal to $5 \cdot 2^{r-4} + r - 4$.

In Chapter 3, a new method of constructing a systematic t-error correcting/all-unidirectional error detecting code, which uses fewer checkbits than any of the previous methods, was proposed. Its decoding algorithm was developed also. Even though this new method shows some improvement it is not yet known whether these codes are optimal. As a matter of fact, there is still some distance between the number of checkbits used in this method and the best known lower bound on the number of checkbits [19]. Thus, further research effort is required to improve this gap.

In Chapter 4, a complete study of Bose-Rao codes, which are the best known single error correcting/all-unidirectional error detecting codes, was done. The maximum Bose-Rao codes for a fixed weight and for all weights was found. Of course, the base group and the group element which make the Bose-Rao code maximal were found, too. An upper bound on the size of a SEC/AUED code, which is derived by Bose, was discussed. This study showed that there is a gap between the maximal Bose-Rao codes and the Bose-bounds. Thus, an improvement is still needed.

In Chapter 5, the nonsystematic SEC/d-UED codes were constructed. Even though the size of the nonsystematic SEC/d-UED code is not yet known to be optimal, it was shown at least close to half of the upper bound which is derived by Bose. Three different methods were proposed for constructing the systematic t-EC/d-UED code. Basically they were constructed in the same way by starting with a systematic parity check code with Hamming distance $2t + 2$ and appending a check symbol which is different for each method. From these constructions, some nice codes were derived, for instance, a systematic SEC/3-UED code with only two more checkbits than the extended Hamming code and a systematic SEC/4-UED code with three more checkbits than the extended Hamming code can be easily constructed from the extended Hamming code. Only the decoding algorithm of one of these methods was developed, but it can be applied to the other two methods. We also derived a lower bound on the number of checkbits for a systematic t-EC/d-UED code. For $t = 1$, the number of checkbits used in the proposed construction is very close to this bound. However, there are still gaps between bounds and the proposed constructions. Thus, further research efforts are needed.

6.2. Totally Self-Checking Checkers

In a fault-tolerant system using error correcting codes the decoder forms the

'hardcore' of the system, i.e. faults in the decoder are not tolerated. Similarly a checker forms the hardcore if we use error detecting codes. Self-checking (SC) circuits are particularly useful in reducing or eliminating this hardcore problem. Generally speaking, a self-checking circuit is a circuit whose output is encoded in an error detecting code, which is a simple code and it is easily observed or checked. Implementation of the circuit as an SC circuit essentially reduces the probability of generating undetectable errors by the circuit due to its internal faults. The other advantage of an SC circuit is that its faults, both transient and permanent, are detected during normal operation. That means it provides concurrent error detection. Moreover, necessary software diagnostic programs are made much simpler or even eliminated. Very important classes of self-checking circuits are self-testing (ST) and totally self-checking (TSC) circuits. Please see [2], [16], [22], [71], [80], [82] for their descriptions, definitions, and conditions.

How to design TSC checkers for the new classes of codes proposed in this thesis is also a topic for future research effort. The TSC checkers must use minimal gate levels for high speed applications and at the same time must have small hardware complexity. Can we implement these TSC checkers using PLA's?

Bibliography

[1] D. A. Anderson, "Design of Self-Checking Digital Networks Using Coding Techniques", Ph.D. Dissertation, University of Illinois, Urbana, IL, Oct. 1971.

[2] D. A. Anderson and G. Metze, "Design of Totally Self-Checking Circuits for m-out-of-n Codes", IEEE Trans. on Computers, vol. C-22, No. 3, pp. 263-269, Mar. 1973.

[3] M. J. Ashjaee and S. M. Reddy, "On Totally Self-Checking Checkers for Separable Codes", IEEE Trans. on Computers, vol. C-26, No. 8, pp. 737-744, Aug. 1977.

[4] J. M. Berger, "A Note on Error Detecting Codes for Asymmetric Channels", Inform. and Control, vol. 4, pp. 68-73, Mar. 1961.

[5] E. R. Berlekamp, *Key Papers in the Development of Coding Theory*, New York: IEEE Press, 1974.

[6] -----, *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.

[7] M. R. Best, A. E. Brouwer, F. J. MacWilliams, A. M. Odlyzko, and N. J. A. Sloane, "Bounds for Binary Codes of Length Less Than 25", IEEE Trans. on Inform. Theory, vol. IT-24, No. 1, pp. 81-93, Jan. 1978.

[8] J. M. Borden, "Optimal Asymmetric Error Detecting Codes", Inform. and Control, vol. 53, No. 1/2, pp. 66-73, Apr./May 1982.

[9] B. Bose, "Theory and Design of Unidirectional Error Codes", Ph.D. Dissertation, Southern Methodist University, Dallas, TX, May 1980.

[10] -----, "On Systematic SEC-MUED Codes", Digest of Papers, 11th International Symp. on Fault-Tolerant Computing, pp. 265-267, Jun. 1981.

[11] -----, "Unidirectional Error Correction/Detection for VLSI Memory", Digest of Papers, 11th International Symp. on Computer Architecture, pp. 242-244, Jun. 1984.

[12] -----, "Two Dimentional ARC Codes", Digest of Papers, 14th International Symp. on Fault-Tolerant Computing, pp. 324-329, Jun. 1984.

[13] -----, "Burst Unidirectional Error Detecting Codes", IEEE Trans. on Computers, vol. C-35, No. 4, pp. 350-353, Apr. 1986.

[14] -----, "On Unordered Codes", to appear.

[15] B. Bose and S. Cunningham, "Systematic and Multiple Error Correcting Asymmetric Codes", IEEE Trans. on Inform. Theory, to appear.

[16] B. Bose and D. J. Lin, "PLA Implementation of k-out-of-n Code TSC Checker", IEEE Trans. on Computers, vol. C-33, No. 6, pp. 583-588, Jun. 1984.

[17] -----, "Systematic Unidirectional Error-Detecting Codes", IEEE Trans. on Computers, vol. C-34, pp. 1026-1032, Nov. 1985.

[18] B. Bose and J. Metzner, "Coding Theory for Fault-Tolerant Systems", Chap. 4 in *Fault-Tolerant Computing* (Theory and Techniques, vol. 1, D. K. Pradhan, Ed.), Englewood Cliffs, NJ: Prentice-Hall, 1986.

[19] B. Bose and D. K. Pradhan, "Optimal Unidirectional Error Detecting/Correcting Codes", IEEE Trans. on Computers, vol. C-31, No. 6, pp. 564-568, Jun. 1982.

[20] B. Bose and T. R. N. Rao, "Theory of Unidirectional Error Correcting/Detecting Codes", IEEE Trans. on Computers, vol. C-31, No. 6, pp. 521-530, Jun. 1982.

[21] -----, "Unidirectional Error Codes for Shift Register Memories", IEEE Trans. on Computers, vol. C-33, No. 6, pp. 575-578, Jun. 1984.

[22] W. C. Carter and P. R. Schneider, "Design of Dynamically Checked Computers", Proc. IFIP Conf., Edinburg, Scotland, vol. 2, pp. 878-883, Aug. 1968.

[23] S. D. Constantin and T. R. N. Rao, "On the Theory of Binary Asymmetric Error Correcting Codes", Inform. and Control, vol. 40, pp. 20-36, Jan. 1979.

[24] R. W. Cook, W. H. Sisson, T. F. Storey, and W. N. Toy, "Design of Self-Checking Microprogram Control", IEEE Trans. on Computers, vol. C-22, No.3, pp. 255-262, Mar. 1973.

[25] C. W. Curtis and I. Reiner, *Representation Theory of Finite Groups and Associative Algebras*, New York: Wiley-Interscience, 1962.

[26] Ph. Delsarte and Ph. Piret, "Bounds and Construction of Binary Asymmetric Error-Correcting Codes", IEEE Trans. on Inform. Theory, vol. IT-27, No. 1, pp. 125-128, Jan. 1981.

[27] -----, "Spectral Enumerators for Certain Additive-Error-Correcting Codes over Integer Alphabets", Inform. and Control, vol. 48, pp. 193-210, Mar. 1981.

[28] H. Dong, "Modified Berger Codes for Detection of Unidirectional Errors", IEEE Trans. on Computers, vol. C-33, No. 6, pp. 572-575, Jun. 1984.

[29] C. V. Freiman, "Optimal Error Detection Codes for Completely Asymmetric Binary Channels", Inform. and Control, vol. 5, pp. 64-71, Mar. 1962.

[30] W. K. Fuchs and J. A. Abraham, "A Unified Approach to Concurrent Error Detection in Highly Structured Logic Arrays", Digest of Papers, 14th International Symp. on Fault-Tolerant Computing, pp. 4-9, Jun. 1984.

[31] W. K. Fuchs, J. A. Abraham, and K. H. Huang, "Concurrent Error Detection in VLSI Interconnection Networks", Digest of Papers, 10th International Symp. on Computer Architecture, pp. 309-315, Jun. 1983.

[32] M. Goto, "Rates of Unidirectional 2-Column Errors Detectable by Arithmetic Codes", Digest of Papers, 10th International Symp. on Fault-Tolerant Computing, pp. 21-25, Oct. 1980.

[33] R. L. Graham and N. J. A. Sloane, "Lower Bounds for Constant Weight Codes", IEEE Trans. on Inform. Theory, vol. IT-26, No. 1, Jan. 1980.

[34] C. Greene and D. J. Kleitman, "Proof Techniques in the Theory of Finite Sets", in *Studies in Combinatorics* (MAA Studies in Math. vol. 17, Gian-Carlo Rota, Ed.), pp. 22-79, Washington, D.C.: MAA, 1978.

[35] R. W. Hamming, "Error Detecting and Error Correcting Codes", Bell Syst. Tech. J., vol. 29, pp. 147-160, Apr. 1950.

[36] -----, *Coding and Information Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1980.

[37] T. Helleseth and T. Kløve, "On Group-Theoretic Codes for Asymmetric Channels", Inform. and Control, vol. 49, pp. 1-9, Apr. 1981.

[38] S. J. Hong and A. M. Patel, "A General Class of Maximal Codes for Computer Applications", IEEE Trans. on Computers, vol. C-21, pp. 1322-1331, Dec. 1972.

[39] M. Y. Hsiao, "A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes", IBM J. Res. Dev., vol. 14, No. 4, pp. 395-401, Jul. 1970.

[40] N. K. Jha and M. B. Vora, "A Systematic Code for Detecting t-Unidirectional Errors", to appear.

[41] W. H. Kim and C. V. Freiman, "Single Error Correcting Codes for Asymmetric Channels", IRE Trans. on Inform Theory, vol. IT-5, pp. 62-66, Jun. 1959.

[42] T. Kløve, "A Class of Constant Weight Codes", Reprot, Dept. of Math., University of Bergen, Nov. 1979.

[43] -----, "Upper Bounds on Codes Correcting Asymmetric Errors", IEEE Trans. on Inform. Theory, vol. IT-27, No. 1, pp. 128-130, Jan. 1981.

[44] -----, "A Lower Bound for A(n, 4, w)", IEEE Trans. on Inform. Theory, vol. IT-27, No. 2, pp. 257-258, Mar. 1981.

[45] D. E. Knuth, "Efficient Balanced Codes", IEEE Trans. on Inform. Theory, vol. IT-32, No. 1, pp. 51-53, Jan. 1986.

[46] S. Lang, *Algebra*, Reading, MA: Addison-Wesley, 1965.

[47] E. L. Leiss, "Data Integrity in Digital Optical Disks", IEEE Trans. on Computers, vol. C-33, No. 9, pp.818-827, Sep. 1984.

[48] S. Lin, *An Introduction to Error-Correcting Codes*, Englewood Cliffs, NJ: Prentice-Hall, 1970.

[49] J. H. van Lint, *Introduction to Coding Theory*, New York: Springer-Verlag, 1982.

[50] C. L. Liu, *Topics in Combinatorial Mathematics* (Notes on Lectures Given at the 1972 MAA Summer Seminar), Williams College, Williamstown, MA, MAA, 1972.

[51] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam, The Netherlands: North-Holland, 1977.

[52] G. P. Mak, J. A. Abraham, and E. S. Davidson, "The Design of PLAs with Concurrent Error Detection", Digest of Papers, 12th International Symp. on Fault-Tolerant Computing, pp. 303-310, Jun. 1982.

[53] M. A. Marouf and A. D. Friedman, "Efficient Design of Self-Checking Checkers for m-out-of-n Codes", IEEE Trans. on Computers, vol. C-27, No. 6, pp. 482-490, Jun. 1978.

[54] -----, "Design of Self-Checking Checkers for Berger Codes", Digest of Papers, 1978 International Symp. on Fault-Tolerant Computing, pp. 179-184, Jun. 1978.

[55] R. J. McEliece, "Comment on a Class of Codes for Asymmetric Channels and a Problem from the Additive Theory of Numbers", IEEE Trans. on Inform. Theory, vol. IT-19, p.137, Jan. 1973.

[56] -----, *The Theory of Information and Coding*, (Encyclopedia of Mathematics and Its Applications, vol. 3), Reading, MA: Addison-Wesley, 1977.

[57] R. J. McEliece and E. R. Rodemich, "The Constantin-Rao Construction for Binary Asymmetric Error-Correcting Codes", Inform,. and Control, vol. 44, pp. 187-196, Jan. 1980.

[58] D. Nikolos, N. Gaitanis, and G. Philokyprou, "t-Error Correcting All Unidirectional Error Detecting Codes Starting from Cyclic AN Codes", Digest of Papers, 14th International Symp. on Fault-Tolerant Computing, pp. 318-323, Jun. 1984.

[59] -----, "Systematic t-Error Correcting/All Unidirectional Error Detecting Codes", IEEE Trans. on Computers, vol. C-35, No. 5, pp. 394-402, May 1986.

[60] B. Parhami and A. Avizienis, "Detection of Storage Errors in Mass Memories Using Low-Cost Arithmetic Error Codes", IEEE Trans. on Computers, vol. C-27, No. 4, pp. 302-308, Apr. 1978.

[61] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, 2nd Edition, Cambridge, MA: MIT Press, 1972.

[62] D. K. Pradhan, "A New Class of Error Correcting/Detecting Codes for Fault Tolerant Computer Applications", IEEE Trans. on Computers, vol. C-29, No. 6, pp. 471-481, Jun. 1980.

[63] D. K. Pradhan and S. M. Reddy, "Fault Tolerant Failsafe Logic Networks", Proc. IEEE COMPCON, pp. 361-363, Mar. 1977.

[64] D. K. Pradhan and J. J. Stiffler, "Error-Correcting Codes and Self-Checking Circuits", IEEE Computer, vol. 13, pp. 27-37, Mar. 1980.

[65] T. R. N. Rao and A. S. Chawla, "Asymmetric Error Correcting Codes for Some LSI Semiconductor Memories", Proc. Annu. Southeastern Symp. on System Theory, pp. 170-171, Mar. 1975.

[66] S. M. Reddy, "A Note on Self-Checking Chedkers", IEEE Trans. on Computers, vol. C-23, pp. 1100-1102, Oct. 1974.

[67] R. L. Rivest and A. Shamir, "How to Re-use A 'Write-Once' Memory", Proc. 14th Annu. Ass. Comput. Mach. Symp. on Theory of Computing, pp. 105-113, May 5-7, 1982.

[68] J. J. Rotman, *An Introduction to Theory of Groups*, Boston, MA: Allyn and Bacon, 1984.

[69] C. E. Shannon, "A Mathematical Theory of Communication", Bell Syst. Tech. J., vol. 27, pp. 379-423 and 623-656, 1948.
Reprinted in: C. E. Shannon and W. Weaver, Eds., A Mathematical Theory of Communication, Urbana, IL: Univ. of Illinois Press, 1963.

[70] J. E. Smith, "On Separable Unordered Codes", IEEE Trans. on Computers, vol. C-33, No. 8, pp. 741-743, Aug. 1984.

[71] J. E. Smith abd G. Metze, "Strongly Fault Secure Logic Networks", IEEE Trans. on Computers, vol. C-27, No. 6, pp. 491-499, Jun. 1978.

[72] E. Sperner, "Ein Satz über Untermengen einer endlichen Menge", Math. Zeitschrift, vol. 27, pp. 544-548, 1928.

[73] R. P. Stanley and M. F. Yoder, "A Study of Varshamov Codes for Asymmetric Channels", Jet Propulsion Lab. Tech. Rep. 32-1526, vol. XIV, pp. 117-122, 1973.

[74] D. Tasar and V. Tasar, "A Study of Intermittent Faults in Digital Computers", Proc. AFIPS Conf., pp. 807-811, 1977.

[75] D. L. Tao, C. R. P. Hartmann, and P. K. Lala, "An Efficient Class of Unidirectional Error Detecting/Correcting Codes", to appear.

[76] R. R. Varshamov, "Some Features of Linear Codes that Correct Asymmetric Errors", Cybern. Control Theory, vol. 9, pp. 538-540, Jan. 1965.

[77] -----, "On the Theory of Asymmetric Codes", Cybern. Control Theory, vol. 10, pp. 901-903, Apr. 1966.

[78] -----, "A Class of Codes for Asymmetric Channels and a Problem from the Additive Theory of Numbers", IEEE Trans. on Inform. Theory, vol. IT-19, pp. 92-95, Jan. 1973.

[79] R. R. Varshamov and G. M. Tenengol'ts, "Correction Code for Single Asymmetrical Errors". Avtomatika i Telemekhanika, vol. 26, No. 2, pp. 288-292, Feb. 1965.

[80] J. F. Wakerly, "Partially Self-Checking Circuits and Their Use in Performing Logical Operations", IEEE Trans. on Computers, vol. C-23, No. 7, pp. 658-667, Jul. 1974.

[81] -----, "Detection of Unidirectional Multiple Errors Using Low-Cost Arithmetic Codes", IEEE Trans. on Computers, vol. C-24, No. 2, pp. 210-212, Feb. 1975.

[82] -----, *Error Detecting Codes, Self-Checking Circuits, and Applications*, Amsterdam, The Netherlands: North-Holland, 1978.

[83] S. L. Wang and A. Avizienis, "The Design of Totally Self-Checking Circuits Using Programmable Logic Arrays", Digest of Papers, 9th International Symp. on Fault-Tolerant Computing, pp. 173-180, Jun. 1979.

[84] C. Y. Wong, W. K. Fuchs, J. A. Abraham, and E. S. Davidson, "The Design of a Microprogram Control Unit with Concurrent Error Detection", Digest of Papers, 13th International Symp. on Fault-Tolerant Computing, pp. 476-483, Jun. 1983.