A STUDY OF BINARY CODES

by

VIRENDRA K. MANAKTALA

B. E. (Electrical),
University of Delhi, India, 1961

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1965

Approved by:

Harley S. Haye
Major Professor

TABLE CONTENTS

INTRODUCTION

A typical communication system, as shown in Fig. 1, employs an encoder to improve the efficiency of channel because an encoded message is less susceptible to noise in the channel. A decoder is employed at the receiving end to recover the original message from the encoded message.

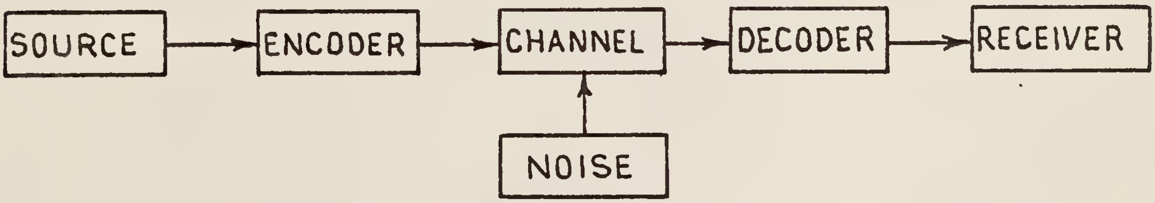
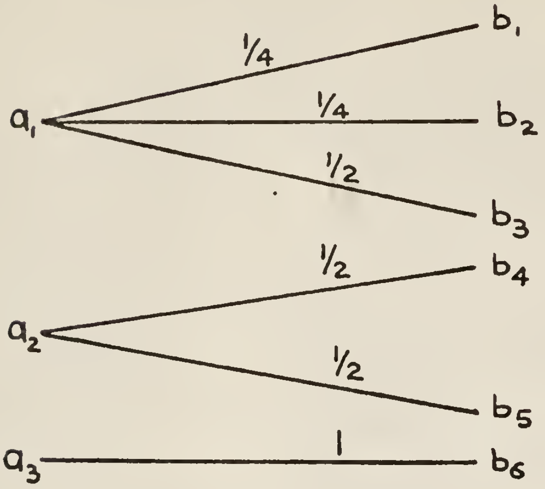


Fig. 1. A typical communication system.

The encoding procedures discussed herein are restricted to binary codes only wherein symbols 0 and 1 form the code alphabet. A letter, symbol, or character is defined as an individual member of an alphabet set, whereas a message or word is a finite sequence of its letters. The length of a word is the number of letters in it.

Encoding or enciphering is a procedure for constructing words using a finite alphabet to represent a given word of the original language on a one-to-one basis. The inverse operation of assigning original words to the encoded words is called decoding or deciphering.

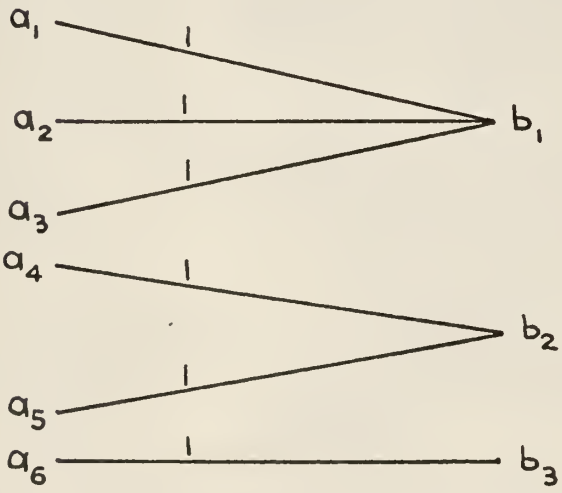
The communication channels may be classified as either noiseless or deterministic. Noiseless channels are characterized by one and only one nonzero element in each column of their channel matrix. A noisy channel is shown in Fig. 2.



$$P = \begin{bmatrix} 1/4 & 1/4 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. A noisy channel and its matrix.

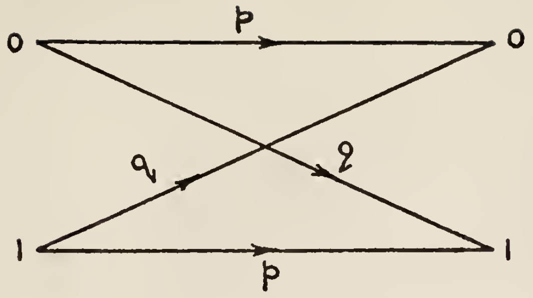
A channel matrix with one and only one nonzero element in each row characterizes a deterministic channel. A deterministic channel is shown in Fig. 3.



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

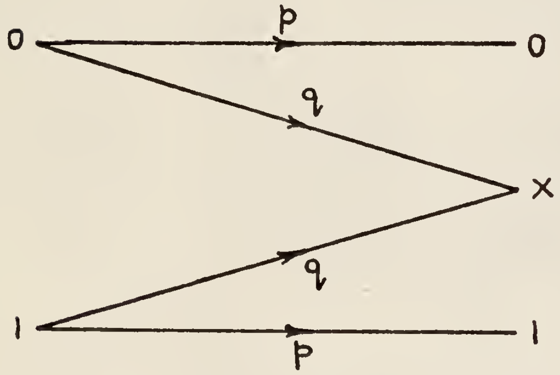
Fig. 3. Deterministic channel and its matrix.

Binary symmetric and binary erasure are two widely discussed channels, and are shown in Figs. 4 and 5, respectively.



$$P = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

Fig. 4. Binary symmetric channel and its matrix.



$$P = \begin{bmatrix} p & 0 & q \\ 0 & p & q \end{bmatrix}$$

Fig. 5. Binary erasure channel and its matrix.

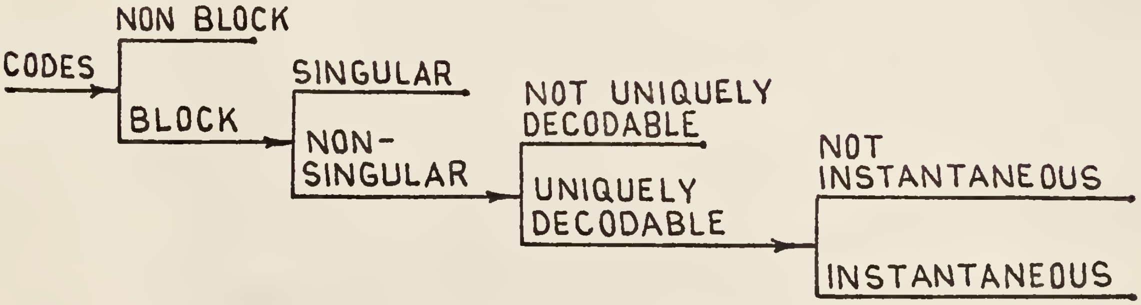


Fig. 6. Classes of codes.

Various classes of codes are summarized (Abramson, 1963) in Fig. 6.

A block code is a code which maps each of the symbols of a source alphabet S into a fixed sequence of symbols of the code alphabet X, as shown in Table 1.

Table 1

Source symbols :	Code
S_1	0
S_2	11
S_3	00
S_4	11

A block code is said to be nonsingular if all the words of the code are distinct.

Table 2. Block code.

Source symbols :	Code (X_i)
S_1	0
S_2	11
S_3	00
S_4	01

A block code is uniquely decodable if, and only if, the n^{th} extension of the code is nonsingular for every finite n. The n^{th} extension of a block code (Table 2) which maps the symbol S_i into code words X_i is the block code which maps the sequence of source symbols ($S_{i1}S_{i2} \ldots S_{in}$) into sequences of code words ($X_{i1}X_{i2} \ldots X_{in}$), as shown in Table 3.

Table 3. Second extension of block code (Table 2).

Source symbols	Code	Source symbols	Code
S_1S_1	00	S_3S_1	000
S_1S_2	011	S_3S_2	0011
S_1S_3	000	S_3S_3	0000
S_1S_4	001	S_3S_4	0001
S_2S_1	110	S_4S_1	010
S_2S_2	1111	S_4S_2	0111
S_2S_3	1100	S_4S_3	0100
S_2S_4	1101	S_4S_4	0101

A uniquely decodable code is said to be instantaneous if it is possible to decode each word in a sequence without reference to succeeding code symbols, as shown below.

Received message	0 0 1 0 1 1 0 1 1 1
Code alphabet	0, 0 1, 0 1 1, 0 1 1 1
Deciphered message	0, 0 1, 0 1 1, 0 1 1 1

Some of the basic binary encoding procedures are Shannon-Fano coding, Shannon binary encoding, Huffman's minimum redundancy codes, and Gilbert-Moore encoding. These codes can neither detect nor correct an error introduced during the transmission of information in a noisy channel. The error-detecting and error-correcting codes are designed to correct certain patterns of errors.

Shannon-Fano Encoding

An ensemble of messages $(X) = (X_1, X_2, \ldots, X_n)$ with probability $(P) = (P_1, P_2, \ldots, P_n)$ are first arranged in the order of decreasing probabilities. This set is then partitioned into two most nearly equiprobable subsets. Then a "0" is assigned to each message in one subset, and "1" to each message in the other subset. The same procedure is repeated until each subset contains only one message as shown in Table 4.

Table 4. Shannon-Fano code.

Message	Probability	Encoded message
X_1	.3	0 0
X_2	.2	0 1
X_3	.2	1 0
X_4	.15	1 1 0
X_5	.1	1 1 1 0
X_6	.05	1 1 1 1

The average length of a code word is defined as $L = \sum P(X_i)n_i$, where n_i is the length of the code word and $P(X_i)$ is the probability of occurrence of message X_i. It is found to be 2.45 for the code in Table 4. The efficiency of coding technique is defined as

$$\eta_i = \frac{H(X)}{\overline{L} \cdot \log D} \qquad (1)$$

where $H(X) = - \sum_i p_i \log p_i$ (2)

D = number of symbols in the code alphabet

The source entropy H(X) can be shown to be 2.408 bits per symbol. The efficiency of the above code is 98.5 per cent.

Shannon's Binary Encoding

An ensemble of messages is written in order of decreasing probabilities and a sequence of 's is computed as shown below.

$$\alpha_1 = 0$$

$$\alpha_2 = P(X_1)$$

$$\alpha_3 = P(X_2) + P(X_1) = P(X_2) + \alpha_2 \qquad (3)$$

where X_1 is the message with highest probability. A set of integers n_i is determined using the inequality

$$P(X_i) \geqslant 2^{-n_i} \text{ for each } P(X_i) \qquad (4)$$

The α's are then expressed in the binary form as shown in Table 5.

Table 5. Shannon's binary code.

Message X_i	Probability $P(X_i)$	α_i	n_i	Code
X_1	.4	0	2	00
X_2	.3	.4	2	01
X_3	.2	.7	3	101
X_4	.1	.9	4	1110

For this case, average length, the source entropy, and the code efficiency are 2.4 bits, 1.846 bits per symbol, and 77 per cent, respectively.

Gilbert-Moore Encoding

A message ensemble is written in the specified order and a set of integers n_i is computed using the following inequality:

$$2^{1-n_i} \leqslant P(X_i) \leqslant 2^{2-n_i} \qquad (5)$$

A sequence of α's is defined below.

$$\alpha_1 = 1/2\ P(X_1)$$

$$\alpha_2 = P(X_1) + 1/2\ P(X_2)$$

$$. \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad .$$

$$\alpha_i = P(X_1) + P(X_2) + \ . \ . \ . + 1/2\ P(X_i) \qquad (6)$$

where X_1 is the first message, X_2 the second, and so on. The α's thus computed are then written as binary digits of length n_i as was done under Shannon's encoding procedure and is shown in Table 6.

Table 6. Gilbert-Moore encoding.

Message X_i	Probability $P(X_i)$	α_i	n_i	Code
X_1	.1	.05	5	00001
X_2	.2	.2	4	0011
X_3	.3	.45	3	011
X_4	.4	.0	3	110

The average length, source entropy, and efficiency for this example are 3.4 bits, 1.846 bits per symbol, and 54.2 per cent, respectively.

Huffman's Minimum Redundancy Codes

$$1 - \text{Efficiency} = \frac{\bar{L} \log D - H(X)}{\bar{L} \log D} \qquad (7)$$

This is the optimum or minimal redundant encoding procedure for a given source which can be coded as shown in Fig. 7. An optimum code satisfies the condition $L(X_{n-1}) = L(X_n)$.

The average length, source entropy, and efficiency for the code in Fig. 7 are 3 bits, 2.972 bits per symbol, and 99.06 per cent, respectively.

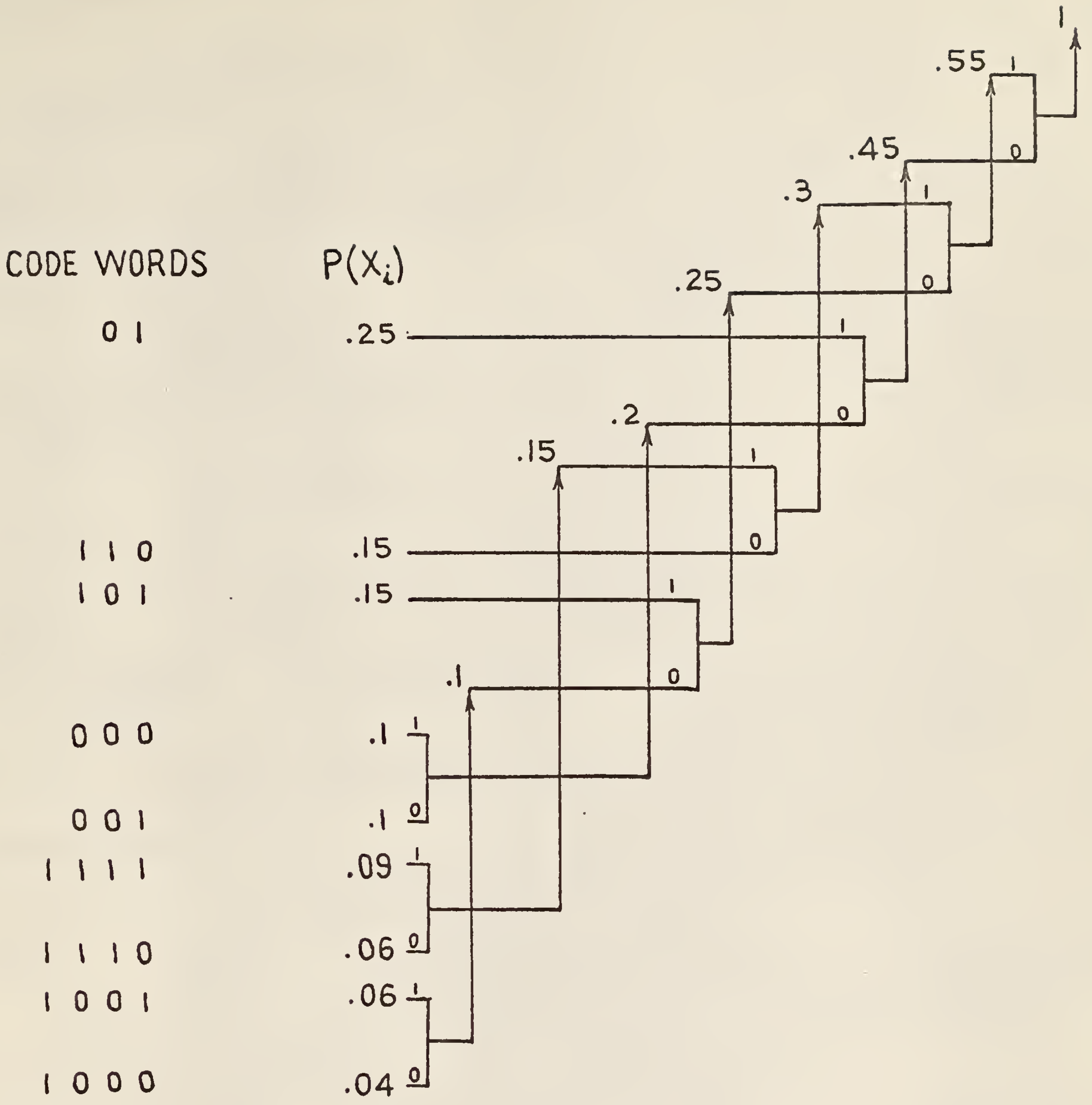


Fig. 7. Huffman's code.

CODING FOR NOISY CHANNEL

Binary codes for the noiseless channels were considered in the last chapter, whereas those for noisy channels are considered here. When a sequence of 0's and 1's is transmitted over a noisy channel, some digits will be received in error and since the source binits are independent, there is no way of detecting erroneous digits. However, if each message digit is transmitted an odd number of times and the most often occurring digits may be selected at the receiver in order to detect the original transmitted sequence. For instance, if each digit in 0010110 is transmitted three times as 000 000 111 000 111 111 111 000 and the received message is 100 001 101 000 110 111 011 010, the original message can be obtained by selecting the digit occurring more frequently in each sequence of three digits. It can be seen that 000 indicates that the transmitted digit is 0, whereas the second sequence 111 indicates 1 as the transmitted digit, and so on. Using this technique, the original message 00101110 is obtained.

Thus it is obvious that in order to detect and/or to correct an error, transmission rate has to be sacrificed, or in other words it is necessary to insert redundant bits in the message at the transmitter. In codes of length n, m message bits are encoded by inserting k redundant digits such as $n = m + k$, and in the above example $m = 1$, $k = 2$, and $n = 3$. Codes with all code words of equal length are usually designated as block codes.

Definitions of some basic terms are summarized below.

Memoryless Channel is a channel in which the probability of error for a received bit is independent of the receiving bits.

Parity Check Digits in a message are used for detection and correction of error.

Even Parity Check in a message n digits long with the first n - 1 information digits and in the nth place, a 0 or 1 is assigned so that the entire message has an even number of 1's.

Odd Parity Check, as in the above case, is an 0 or 1 in the nth place, such that the entire message has an odd number of 1's.

Weight of a Code Word is the number of 1's in the word.

Group Code is a binary code having the group property (see Appendix).

Systematic Code is a block code of length n, with m information digits and k + n - m parity check digits.

Linear Code is a mathematical term for n-ary (binary, ternary, etc.) code having a specific property. For binary codes, terms like linear codes and group codes are synonymous.

Hamming Codes

The first error correcting and detecting codes are due to Hamming (1950). Some details of these codes follow.

Single Error Detecting Code (SED) is, for instance, an n bits long code with even parity check. Any single error would result in an odd number of 1's in the received code words. All other errors will go undetected.

Single Error Correction Codes (SEC) can correct any single error occurring within a code word. Consider a code n binits long, m of which are information digits and the remaining k = n - m are parity check digits. The values of the check digits are determined in the encoding operation by even parity checks over selected information places. Parity check rules are assigned as shown later, and then the parity checks are applied in such a way that every time a parity check agrees with the value observed in the corresponding check digit, a "0" is recorded and a "1" is recorded otherwise. This sequence of "0's" and "1's", k digit long written from right to left is a binary number giving the position of the error. Since the number is required to give the position of single error in the code, any position having a "1" on the right of its binary representation must cause the parity check to fail. So the first parity check must be over positions which have "1" as the first digit in binary numbers, such as 1, 3, 5, 7, 9, ..., etc. Similarly, the second parity check should be over the position numbers whose binary representation has "1" as their second digits, namely, 2, 3, 6, 7, ..., etc. Similarly, positions for the third parity check would be 4, 5, 6, 7, etc. See Table 7.

An illustration of Hamming's SEC code for n = 7 is discussed here. Information positions in the code shown above are 3, 5, 6, 7. If the corresponding information digits are 1 0 1 1, the

Table 7

Parity check number	Location of check digit	Positions checked
1	1	1,3,5,7,9, ...
2	2	2,3,6,7,10,11, ...
3	4	4,5,6,7,12, ...
4	5	8,9,10,11,12 ...

complete code word would be 0110011 using even party checks, there are 4 information positions, $(2)^4$ = 16 different sequences of the code words are possible. Suppose the received word is 011011. Using the procedure described above,when first parity check is applied over positions 1, 3, 5, 7, it is found that the digit in position 1 should be a 1, but there is a zero in that position. So the first digit of checking number is 1. Similarly, the digit in second position agrees and the digit in fourth position disagrees, so the checking number is 101. Hence the error is in fifth position as 101 is the binary representation of 5.

Single Error Correcting and Double Error Detecting Codes can be obtained from single error correction codes by simply adding an additional digit in the end, which is an even parity check over all previous digits. The operation of this code is summarized in Table 8.

Hamming introduced a geometrical model to specify the requirements for codes which can either detect or correct more than two errors. This consists of identifying sequences of 0's and 1's in each code word with a point in n-dimensional space. For large n, this concept is difficult to visualize but

Table 8

Type of error	Correction procedure
No error	All parity checks are satisfied and checking number is 000.
Single error	Last parity check fails. Checking number indicates the position of error. If checking number is zero, it means an error in the last check position.
Double error	Last parity check is satisfied and the checking number is obtained but gives no information about the position of error.
More than two errors	No useful information is obtained and if the resulting number of errors is odd, checking number might create an additional error.

the case of n = 3 can be used to explain the basic concept. For n = 3, $(2)^3 = 8$ words can be associated with the corners of a cube as shown in Fig. 8. The distance D(U, V) between two

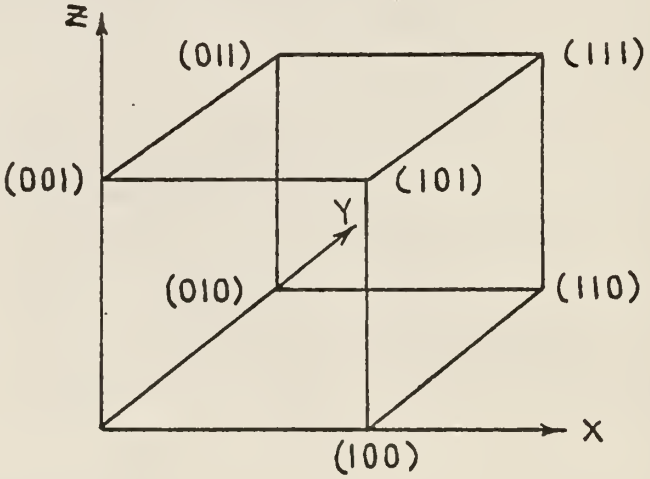


Fig. 8. Three-dimensional cube.

points, U and V, is the number of positions in which they differ. If U = 100 and V = 011, then D(U, V) = 3.

This shows that in order to make a transition from 100 to 011, or vice versa, three sides of the cube must be traversed. If all the points are used as code words, the error cannot be detected. Now if the code words having a minimum distance of 2 units are chosen, a single error can be detected because it will cause the code point to be moved in only one direction to a point which is not defined as a code word. One such set of symbols would be 000, 011, 101, 110.

Also if the minimum distance between code words is at least 3, a single error will leave the displaced point nearer to correct point than to any other code point, and therefore a single error can be corrected. The above discussion is summarized in Table 9.

Table 9. Hamming's error detecting and correcting codes.

Minimum distance	Correction procedure	Detected	Corrected
1	No error detection or correction	0	0
2	SED	1	0
3	SEC	1	1
4	SEC-DED	2	1
5	DEC or (DEC-DEC)	2	2
6	DEC-TED	3	2

Slepian Group Codes

Slepian (1956) generalized the codes obtained by Hamming and Reed-Muller. Actually Hamming and Reed-Muller codes form a subclass of a larger class of codes called Group Codes.

The encoding scheme for these codes is simple because last K positions in the code word are assigned parity checks and the decoder, which is a maximum likelihood detector, is optimum theoretically.

The mathematical definition of group is given in the appendix. A collection of binary words is said to form a group if the product of any two members is also a member of this collection, and this collection contains the identity element. As an example, the 2^n possible sequences of n bits form a binary group. This group is denoted by B_n and contains 2^n words or elements. An n-place group code is defined to be a collection of $2^m (m < n)$ binit. Code words that form a group as defined above. All such group codes are subgroups of B_n which contains all 2^n possible sequences of n binits.

According to Slepian there are exactly

$$N(n,m) = \frac{(2^n - 2^0)(2^n - 2^1) \ldots (2^n - 2^{m-1})}{(2^m - 2^0)(2^m - 2^1) \ldots (2^m - 2^{m-1})} \qquad (8)$$

different subgroups of B_n having 2^m elements. And these N(n,m) possible codes are tabulated by Slepian (1956) and many examples of such codes are also given.

After choosing a (n,m) code and 2^m words, the information is transmitted by selecting blocks of m message digits and associating these in a one-to-one manner with the 2^m code words.

Since the channel is noisy, some digits in the received code word will be in error. These errors are then corrected knowing that transmitted words formed a group.

Slepian has suggested an optimum decoding method which uses a standard array of code words, which is obtained as follows. The words of a specific code word $A_1 = I = 000 .. 0$, $A_2, A_3 \ldots A_\mu$ are placed in the top row and the words $S_2, S_3 \ldots S_v$ are chosen such that S_2 is any code word not contained in the first row. S_3 is any code word not contained in the first two rows and so on. The whole array then can be developed as shown below.

$$B_n = \begin{array}{llll} I & A_2 & A_3 & \qquad\quad A_\mu \\ S_2 & S_2A_2 & S_2A_3 \; \ldots\ldots & S_2A_\mu \\ S_3 & S_3A_2 & S_3A_3 \; \ldots\ldots & S_3A_\mu \\ \vdots & \vdots & \vdots & \quad\vdots \\ S_v & S_vA_2 & S_vA_3 \; \ldots\ldots & S_vA_\mu \end{array} \qquad (9)$$

All the rows except the first are called Cosets and the first words $S_2, S_3, \ldots, S_v$ are called Coset Leaders. The weight W_{ij} of an element in the array is the number of nonzero digits in the n binit word located in i^{th} row and j^{th} column. If the coset leaders are chosen as the elements with the minimum weight, such an array is called Standard Array. Peterson (1961) gives several theorems using the properties of standard array coset and coset leaders. Slepian (1956) and Prange (1957) offered a theorem using properties of coset and coset leader for detection of group codes. The decoding process is illustrated with a specific example as given below.

A standard array for (4,2) code 0000, 1100, 0011, 1111 is as follows:

0000	1100	0011	1111
0001	1101	0010	1110
0100	1000	0111	1011
0110	1010	0101	1001

It is formed by the method illustrated in the preceding paragraph. Note that many standard arrays can be formed by choosing different coset leaders having the same weight. Considering a BSC, when a word, say A, is sent, the word received can be any element of B_n. If the received word lies in the i^{th} column, the detector will show that word A_i has been transmitted. In the above example, 0111 will be produced as 0011, and 1010 will be produced as 1100. Moreover, considering the fact that any word in a standard array is at least as close to the code words at the top of its column as it is to any other transmitted code word, this scheme gives maximum likelihood detection that is the detected symbol is the one most likely to have been transmitted. The only drawback of this decoding scheme is that it requires knowledge of all 2^n possible words. Hence the detection equipment grows exponentially with the length of the code word. To overcome this disadvantage, a simple encoding procedure for the generation of code words by parity checks is given in the following example. Definitions of Systematic Code and Equivalent Code are also given below.

In a Systematic Code, all code words have the same information digit locations and same parity check digit locations.

Two group codes are Equivalent if one can be obtained from the other by permitting the digit locations. The generation of groups codes by parity check is explained in the following example.

Consider the same (4,2) code given by 0000, 1100, 0011, 1111. Positions 2 and 3 are assumed to be information positions. If a code word is denoted as $X_1X_2X_3X_4$, the parity check rules are determined by solving the simultaneous equations

$$X_1 = A_1X_2 \oplus A_2X_3 \tag{10}$$

$$X_4 = A_3X_2 \oplus A_4X_3 \tag{11}$$

(Note that $\oplus$ means addition modulo 2.) Substituting values from third and fourth code words,

$$\begin{aligned} 1 &= A_1 \cdot 0 \oplus A_2 1 & \qquad A_2 &= 1 \\ 1 &= A_1 \cdot 1 \oplus A_2 1 & A_1 &= 0 \end{aligned}$$

$$\begin{aligned} 1 &= A_3 \cdot 0 \oplus A_4 1 & \qquad A_4 &= 1 \\ 1 &= A_3 \cdot 1 \oplus A_4 1 & A_3 &= 0 \end{aligned}$$

so $A_2 = A_4 = 1$ and $A_1 = A_3 = 0$.

Thus the parity check rules are $X_1 = X_3$ and $X_4 = X_3$.

Slepian (1956) has given two important results involving systematic and equivalent code.

1. Every group code is a systematic code and vice versa.

2. Every (n,m) code is equivalent to a (n,m) code having first m places as information digits and remaining $k = n - m$ places as parity check over the first m places. The general expression for k check digits then becomes

$$X_i = \sum_{j=1}^{m} v_{ij}X_j \text{ where } i = m + 1, \ldots, n \tag{12}$$

(Summation is modulo 2 with the multiplication rules being $0 \cdot 1 = 1 \cdot 0 = 0 \cdot 0 = 0$, and $1 \cdot 1 = 1$.) Slepian also lists suitable parity check rules for optimum results. For instance, consider a (6,3) code. In this case number of words = $(2)^3 = 8$ and parity check rules are:

$$X_4 = X_1 \oplus X_2 \qquad (13)$$

$$X_5 = X_1 \oplus X_3 \qquad (14)$$

$$X_6 = X_2 \oplus X_3 \qquad (15)$$

resulting in Table 10.

Table 10

Code words	Information digits	Parity check
1	000	0 0 0
2	001	0 1 1
3	010	1 0 1
4	011	1 1 0
5	100	1 1 0
6	101	1 0 1
7	110	0 1 1
8	111	0 0 0

Slepian has described a method of maximum likelihood detection by means of parity checks. In this method for a specific (n,m) code, parity check rules are applied to any code word, say T, and check digits resulting from the i^{th} parity check are compared with the digit in 2nd position of T. If these agree, a "0" is assigned; otherwise a "1" is assigned. In this way a parity check sequence denoted by R(T) is thus formed by writing "1" and "0" from left to right.

If the message received is 101001 = T, and the first three bits are information digits, namely, 101, then the parity check word for this code is 001, while (from Table 4) it should be 101. Since 4th parity check is different and 5th and 6th parity checks are the same, the checking number is 100 = R(T), and therefore the error is in the 4th position.

Using the parity check rules of the last example,

Let T = (101001)

$X_1 + X_2 = X_4$ so assign a "1"

$X_1 + X_3 = X_5$ so assign a 0

$X_2 + X_3 = X_6$ so assign a "0"

R(T) = (100)

Slepian has proved the following theorem using this definition of R(T). If I, A_2, A_3, ..., A_μ is a (n,m) code and consider B_n to be developed in this standard array by this code. Let R(T) be the parity check sequence for a word T which has been formed in accordance with the parity check rules of the specified code. Then $R(T_1) = R(T_2)$ if and only if T_1 and T_2 lie in the same row of the standard array.

Consider (4,2) code

0000	1011	0101	1110
0010	1001	0111	1100
0100	1111	0001	1010
1000	0011	1101	0110

Parity check rules are $X_3 = X_1$ and $X_4 = X_1 \oplus X_2$.

By definition all words in the first row satisfy parity check giving a parity check sequence of 00. In the second row, every word fails the first parity check and satisfies the second check. Therefore the parity check sequence for this row is 10. Similarly, parity check sequences for third and fourth row are 01 and 11, respectively. So one can establish the following correspondence between the parity check sequence and coset leaders.

$$00 \longrightarrow S_1 = 0000$$
$$10 \longrightarrow S_2 = 0010$$
$$01 \longrightarrow S_3 = 0100$$
$$11 \longrightarrow S_4 = 0001$$

Maximum likelihood detection is now obtained as described in the following paragraph.

Parity check sequence R(T) of the received code word T is formed, which in turn identifies the coset leader, say S_i. The product S_i T is produced as the detector output. Suppose (0001) of (4,2) code is received. R(0001) = 01 wherein corresponds to S_3 = 0100. The detected word is therefore (0100) (0001) = 0101.

Some of the recent and advanced block coding techniques (Peterson, 1961) are discussed in this section. Some pertinent definitions are given here to supplement the material in the appendix.

Bose-Chaudhuri Codes

These cyclic codes are defined in terms of the roots of a generator polynomial. A generator polynomial g(x) is defined as the monic polynomial of the smallest degree, such that g(x) is in an ideal I (see appendix for definition of an Ideal).

Assume the symbols to be elements of Galois field of q elements, i.e., GF(q). Let m be an integer and L be any element of GF(q^m). Then the code consisting of all vectors (f(x)) over GF(q) for which α^{m_0}, α^{m_0+1}, ..., α^{m_0+d-2} are roots of f(x) is a Bose-Chaudhuri code. For the binary case q = 2 and let

$m_0 = 1$ and $d = 2^{t+1}$ where t is any integer. $(f(x))$ is a code vector if and only if $\alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ are roots of $f(x)$. The minimum function of any element B in the extension field (see appendix) is defined as the monic polynomial of the smallest degree with the coefficients in the ground field F. Let $m_1(x)$ denote the minimum function of 1^i where 1 is a primitive element of $GF(q^m)$ then $\alpha^1, \alpha^2, \alpha^4, \ldots,$ are roots of $m_1(x) = m_2(x) \ldots m_4(x) = m_8(x) \ldots, \alpha^3, \alpha^6, \alpha^{12} \ldots$ are roots of $m_3(x) = m_6(x)$, and α^5, α^{10} are roots of $\ldots m_5(x) = m_{10}(x) \ldots$.

Hence in other words, all the even powers of α are a root of the same minimum function of some previous odd powers of α, i.e., $\alpha^2, \alpha^4, \alpha^8, \ldots,$ are roots of $m_1(x)$; α^6, α^{12} are roots of $m_3(x)$, and α^{10} is a root of $m_5(x)$. Therefore it may be said that $(f(x))$ is a code vector if and only if $\alpha, \alpha^3, \ldots, \alpha^{2t-1}$ are roots of $f(x)$ where t is any integer. Therefore the generator of code is the Least Common Multiple (L C M) of $(m_1(x)\ m_3(x)\ \ldots\ m_{2t-1}(x))$.

The following example illustrates the procedure for constructing a Bose-Chaudhuri Code. Let $q = 2$ and α be a primitive element of $GF(2^4)$. Then $\alpha^{15} = 1$.

$\alpha^1, \alpha^2, \alpha^4, \alpha^8$ are roots of $m_1(x) = m_2(x) = m_4(x) = m_8(x)$ degree 4
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^9$ are roots of $m_3(x) = m_6(x)$ degree 4
α^5, α^{10} are roots of $m_5(x) = m_{10}(x)$ degree 2

Therefore $g(x) = m_1(x)m_3(x)m_5(x)$ (16)

If α is a root of $x^4 + x + 1$ (see Table 11 in Appendix), then $m_1(x) = x^4 + x + 1$. $m_3(x)$ and $m_5(x)$ are calculated as shown.

Let $m_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + x^4$ (17)

Substitute $\alpha^3 = (0001)$ for x

$\alpha^6 = (0011)$ for x^2

$\alpha^9 = (0101)$ for x^3

$\alpha^{12} = (1111)$ for x^4

Then

$$a_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + a_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + a_3 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + a_4 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 0$$

or

$$\begin{aligned} a_0 + 1 &= 0 & \text{Therefore} \quad a_0 &= 1 \\ a_3 + 1 &= 0 & a_3 &= 1 \\ a_2 + 1 &= 0 & a_2 &= 1 \\ a_1 + a_2 + a_3 + 1 &= 0 & a_1 &= 1 \end{aligned}$$

Therefore $m_3(x) = 1 + x + x^2 + x^3 + x^4$ (18)

Similarly $m_5(x)$ can be found to be $= (x^2 + x + 1)$

Therefore

$$\begin{aligned} g(x) &= (1 + x + x^4)(1 + x + x^2 + x^3 + x^4) \\ &\quad (1 + x + x^2) \\ &= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10} \end{aligned} \tag{19}$$

The multiplication process is illustrated in the Appendix. The degree of $g(x) = 10$ and therefore the dimension of the code is $15 - 10 = 5$, which corresponds to the information digits. The elements are

$$\begin{aligned} (g(x)) &= 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\ (xg(x)) &= 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \\ (x^2g(x)) &= 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0 \\ (x^3g(x)) &= 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0 \\ (x^4g(x)) &= 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0 \end{aligned}$$

The generator matrix is

$$G = \begin{bmatrix} 1&1&1&0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&1&1&1&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&1&1&1&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&1&1&1&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&1&1&1&0&1&1&0&0&1&0&1&0 \end{bmatrix} \tag{20}$$

Bose and Chaudhuri have summarized all the codes generated by primitive elements of order less than 2^8 (Peterson, 1961).

Reed-Muller Codes

These codes are a subclass of Slepian codes and cover a wide range of rate and minimum distance. These are different from Slepian codes in that a specific procedure is available for determining a set of code words when the following relations hold:

$$n = 2^m \tag{21}$$

$$k = 1 + \binom{m}{1} + \ldots + \binom{m}{r} \tag{22}$$

$$n - k = 1 + \binom{m}{1} + \ldots + \binom{m}{m-r-1} \tag{23}$$

$$d = 2^{m-r} = \text{minimum weight} \tag{24}$$

$$r = \text{order of the code}$$

Consider the set of vectors V_0, V_1, V_2, ..., V_m, over the field of two elements. Let V be the vector whose 2^m components are all 1's and V_1, V_2, ..., V_m be the rows of a matrix that has all possible m-tuples, i.e., 1.2 .. 2^m as columns. Define the vector product of two vectors as follows.

$$U = (a_1, a_2, a_3, \ldots, a_n) \tag{25}$$

$$V = (b_1, b_2, b_3, \ldots, b_n) \tag{26}$$

$$UV = (a_1b_1, a_2b_2, a_3b_3, \ldots, a_nb_n) \tag{27}$$

The collection of vectors formed by multiplying vectors V_i two at a time, three at a time, ... m at a time, gives us the basis for Reed-Muller codes.

Consider the case of $m = 3$, $r = 1$. Then length of code $n = 2^3 = 8$. $k = 1 + \binom{3}{1} = 4$.

This is an (8,4) code.

$$\begin{aligned}
V_0 &= (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)\\
V_3 &= (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)\\
V_2 &= (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)\\
V_1 &= (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)\\
V_3V_2 &= (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)\\
V_3V_1 &= (0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)\\
V_2V_1 &= (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)\\
V_3V_2V_1 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)
\end{aligned}$$

These eight vectors are then the basis of a first-order Reed-Muller code. In a way these are equivalent to the generator matrix as in the Bose-Chaudhuri case.

Fire Codes

These codes with symbols from GF(q) are best defined in terms of the generator polynomial.

$$g(x) = p(x)(x^c - 1) \qquad (28)$$

where p(x) is an irreducible polynomial of degree m over GF(q) whose roots have order c, and is not divisible by c. The number of parity checks is $(c + m)$, and therefore the number of information symbols $k = n - c - m$.

For instance, consider the fire code generated by

$(x^7 + x^2 + 1)(x^{16} + 1)$. In this, $m = 7$, $c = 16$, order of roots $= 2^7 - 1 = 127$, and $n = 16 \times 127 = 2032$. Parity check symbols $= c + m = 7 + 16 = 23$. Information symbols $= 2032 - 23 = 2009$.

Fire codes are capable of correcting any single burst of length $b \leqslant m$ or less, and of simultaneously detecting any burst of length $d \geqslant b$ or less if $c \geqslant b + d - 1$. In the example above, the code can correct all bursts of length 7 or less and simultaneously detect all bursts of length 10 or less. It can detect any burst of length 23 or less and the combination of two bursts in which the shorter has length no greater than 7 and the longer has length no greater than 17.

The procedure for constructing a code is illustrated below.

Let
$$g(x) = (x^3 + x^2 + 1)(x^5 + 1) \tag{29}$$

Then $m = 3$, $c = 5$. Order of roots of $x^3 + x^2 + 1 = 2^3 - 1 = 7$. Therefore length $n = 7 \times 5 = 35$. Parity check symbols $= c + m = 5 + 3 = 8$. Information symbols $= 35 - 8 = 27$.

$$g(x) = 1 + x^2 + x^3 + x^5 + x^7 + x^8$$

35th digit

$$
\begin{aligned}
\big(g(x)\big) &= (1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ \ldots\ldots\ldots\ 0) \\
\big(xg(x)\big) &= (0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ \ldots\ldots\ldots\ 0) \\
\big(x^2g(x)\big) &= (0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ \ldots\ldots\ldots\ 0) \\
&\ \ \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\big(x^7g(x)\big) &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \ldots 0)
\end{aligned}
$$

Therefore the generator matrix G is then

$$G = \begin{array}{lr} & \text{35th digit} \\ \left[\begin{array}{l} 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0 \ldots\ldots\ldots \\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0 \ldots\ldots\ldots \\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \ldots\ldots\ldots \\ \vdots \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0 \ldots\ldots \end{array}\right. & \left.\begin{array}{r} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \end{array} \qquad (30)$$

This fire code is capable of correcting a burst of length 3 or less and simultaneously detecting a burst of length 3 or less. For detection alone it can detect a burst of length no greater than 8 and a combination of two bursts in which the shorter burst is no greater than 3 and the longer burst is no greater than 6.

Reed-Solomon Codes

These codes are capable of correcting more than one burst of error. Peterson defines this as a code with symbols in GF(q^r) that corrects all bursts of t or fewer errors on a channel. The elements of GF(q^r) are encoded into blocks of r symbols of GF(q). Then the burst of length $(t - 1)r + 1$ or less can affect at most t excessive blocks, and hence can be corrected.

For example, consider a Reed-Solomon code with symbols from GF(2^7) and $t = 4$, $r = 7$, $t = 4$. This could correct all bursts of length $(t - 1)r + 1$, i.e., $(4 - 1)7 + 1 = 22$ or less. The number of blocks required $= 2 \times 4 = 8$, the number of parity checks $= 8 \times 7 = 56$, and the length of code is $7 \times 2^7 - 1 = 7 \times 127 = 889$. The length of Fire code to correct the same burst of length, i.e., 22, would be $(2^{22} - 1) \times 43 = 160$ million binary digits. Thus it is clear that the length of Reed-Solomon code compared with the Fire code will be much less for correcting the burst of the same length.

Summary

Hancock (1962) has summarized the relationship between the various encoding procedures considered in the previous sections by means of a block diagram, shown in Fig. 9. He compares the various encoding procedures for the noisy channels giving the advantages and disadvantages of each.

BIT LOSS AND GAIN CORRECTION CODE

The first single bit loss or gain correction code has been suggested by Sellers (1962). In digital communication systems a message may be received with a different number of bits than those in the transmitted message. In other words, a message may gain or lose bits due to the loss of synchronization between the transmitter and the receiver among other causes. The synchronization systems may be classified, (Wier, 1961), as synchronous and asynchronous. In the synchronous system, each data bit takes the same time for transmission, while in asynchronous system data bits may take unequal times for transmission. In the synchronous system a clock at the transmitter times the bits and the blocks prior to transmission and thus determines the rate of the data source. At the receiver end a clock or an

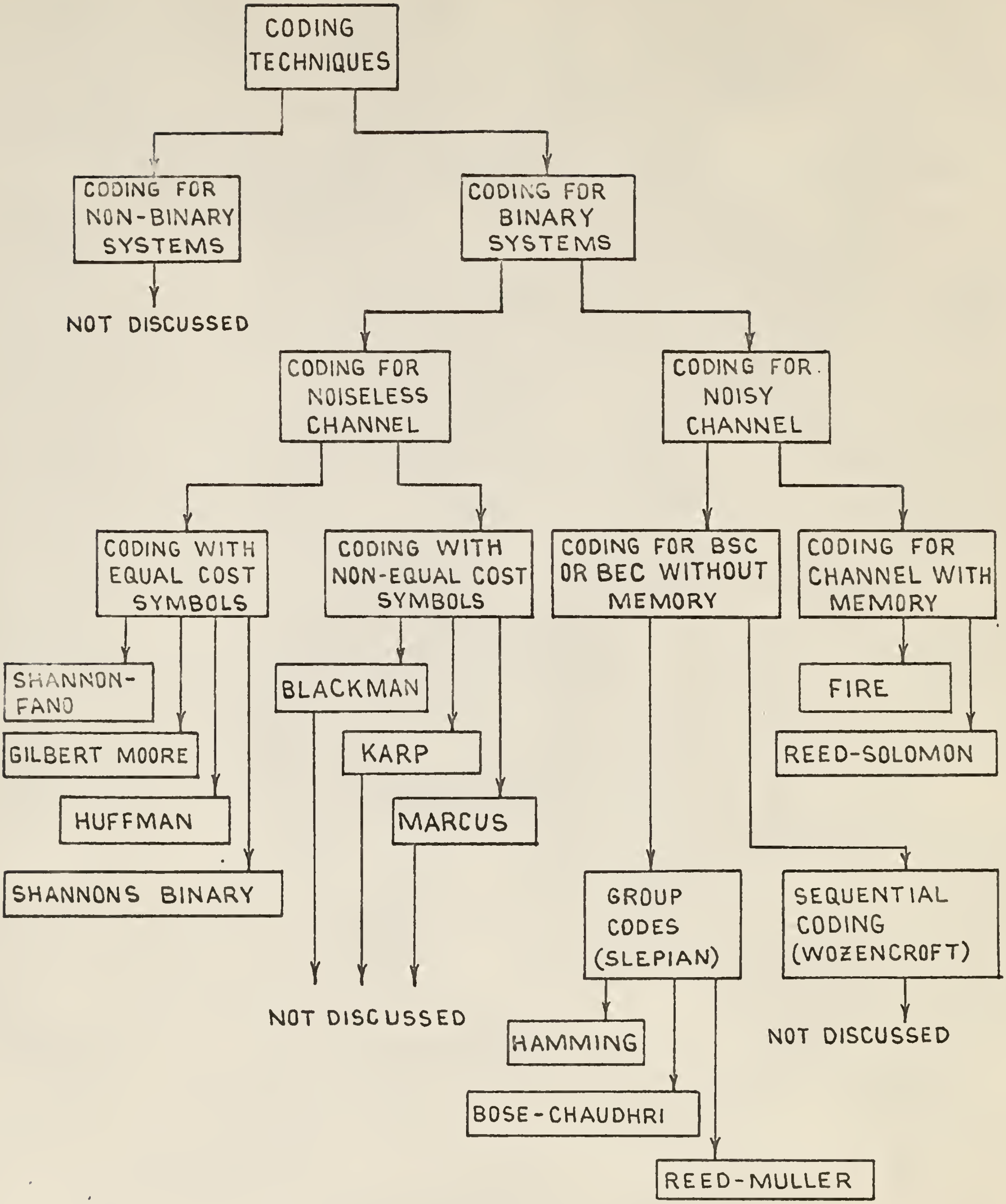

CODING TECHNIQUES
CODING FOR NON-BINARY SYSTEMS
CODING FOR BINARY SYSTEMS
NOT DISCUSSED
CODING FOR NOISELESS CHANNEL
CODING FOR NOISY CHANNEL
CODING WITH EQUAL COST SYMBOLS
CODING WITH NON-EQUAL COST SYMBOLS
CODING FOR BSC OR BEC WITHOUT MEMORY
CODING FOR CHANNEL WITH MEMORY
SHANNON-FANO
GILBERT MOORE
HUFFMAN
SHANNONS BINARY
BLACKMAN
KARP
MARCUS
FIRE
REED-SOLOMON
NOT DISCUSSED
GROUP CODES (SLEPIAN)
SEQUENTIAL CODING (WOZENCROFT)
HAMMING
BOSE-CHAUDHRI
REED-MULLER
NOT DISCUSSED

Fig. 9. Summary of relationship between various encoding procedures.

oscillator pulled into proper phase, or a shock-excited ringing circuit of proper phase is used to record each data bit once and only once.

When a message is received with more or fewer bits than were transmitted, it may be corrected by approximately locating the error and removing or inserting bits, respectively. The approximate location of the bit loss or gain is found by the use of a special character which is inserted into the code at periodic intervals. The special character has the property that when shifted left or right or when a bit is gained or lost in it, it gives an indication of what has happened and thus determines the corrective action.

Special character 001 will be used for the following two examples which indicate the procedure for the case of a single bit loss or gain. Let the encoded message be 001a_2, a_{2j-1}, ..., a_{j+1}001 a_j ... a_2a_1, where a_1 is the first bit to be sent into the channel. On receipt of a count j bits from a_1, one examines for the special character.

The length is corrected accordingly if a bit is lost or gained. It is possible that either an incorrect bit may be inserted or removed, or a bit may be inserted or removed from the wrong position. This results in a burst of errors, which is then corrected by a burst error correcting code.

The size of burst of errors is limited by the fact that in order to correct any burst of length b or less, a linear code must have at least 2b parity check symbols (Peterson, 1961).

It can be generalized as follows.

J = length of block	b = length of burst possible
1	$1 = \frac{J+1}{2}$
2	$2 = J/2 + 1$, or $1 = J - 1$
3	$3 = J$, or $2 = \frac{J+1}{2}$, or $1 = \frac{J-1}{2}$
4	$4 = J$, or $3 = \frac{J}{2} + 1$, or $2 = \frac{J}{2}$, or $1 = \frac{J-2}{2}$
5	$5 = J$, or $4 = \frac{J-1}{2}$ or $3 = \frac{J+1}{2}$, or $2 = \frac{J-1}{2}$ or $1 = \frac{J-3}{2}$

In the case of even J the only term occurring for all even values is J/2 + 1 and it is also the maximum length of burst which can be corrected, according to the theorem stated above. So

$$b \leqslant J/2 + 1 \tag{31}$$

Similarly, for odd J the only term occurring for all odd values is $\frac{J+1}{2}$ and it is also the maximum length of burst of errors which can be corrected. So

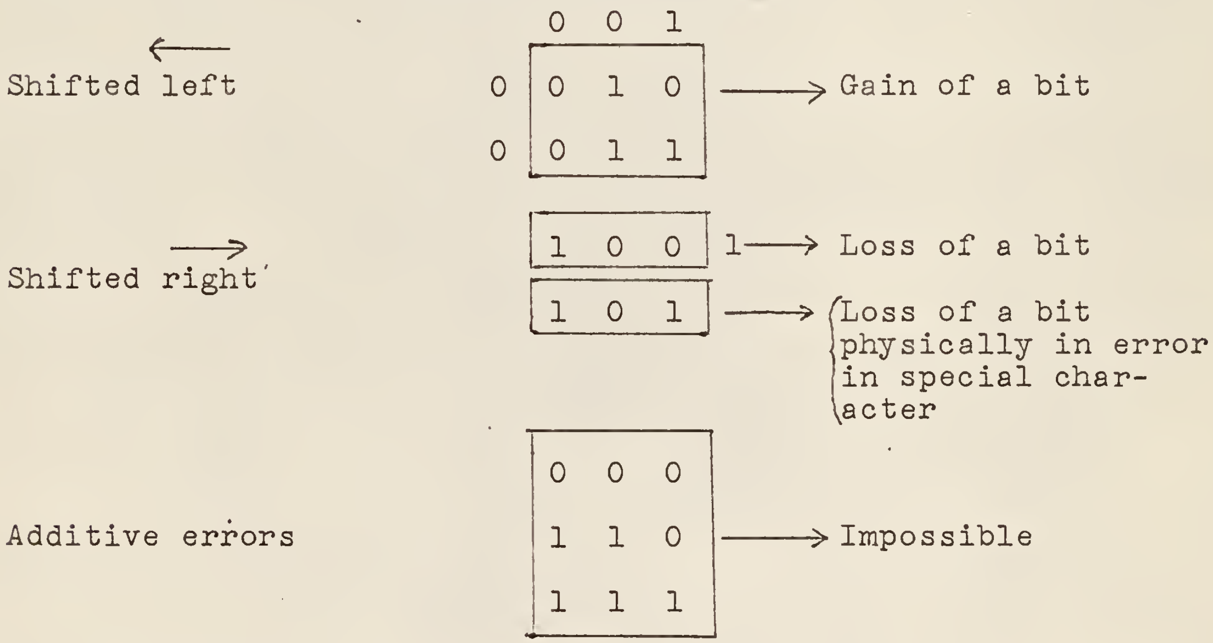
$$b \leqslant \frac{J+1}{2} \tag{32}$$

An additive error is defined as a change of "1" to "0" or vice versa. When the length correction is applied by examining just one special character without reference to any other special character, the process is called <u>independent decoding</u>. This is not capable of correcting an additive error because if such an error is allowed, the error may occur in the special character and thus result in a wrong correction procedure. On the other hand, if dependent decoding which includes checking two consecutive special characters and the information contained in these characters is used for taking the corrective action, the additive errors can be corrected. In this case also

$$b \leqslant J/2 + 1 \quad \text{for } j \text{ even}$$

$$b \leqslant \frac{J + 1}{2} \quad \text{for } j \text{ odd}$$

First case Independent decoding



Consider an example of J = 5. Let the encoded message be

$$1\;1\;1\;1\;\underbrace{0\;0\;1}\;1\;1\;0\;0\;1\;\underbrace{0\;0\;1}\;1\;1\;0\;1\;1$$

and the message received be

$$1\;1\;0\;\underbrace{0\;1\;1}\;1\;0\;0\;1\;0\;\underbrace{0\;1\;1}\;1\;0\;1\;1\;1$$

whereas the special character is received as 011.

There is gain of one bit and so one bit has to be removed. The requirement is to remove that bit such that if it causes a burst of errors, it should not exceed the permissible length of burst of error, in this particular case $= \dfrac{5 + 1}{2} = 3$.

Hagelberger (1959) defines the length of a burst as the length of a binary word which describes the error pattern and is formed by putting a "1" for each erroneous digit and a "0" for each correct digit. One essential requirement is that the first and last digit of such a binary word be "1" because there is no point in including correct digits which are outside the burst.

So if a_1 is removed and it may be incorrect and if a_4 and a_5 are also in error, we will get a burst of length 5 which cannot be corrected. If a_3 is removed no matter where the burst started in the block, the burst will never exceed the length 3 and so can always be corrected. It is shown below that for all odd J and for a single bit gain, the bit to be removed would be $a_{(J+1/2)}$ and similarly for even J, $a_{(J/2+1)}$. Similarly, if a loss of a bit occurs, then the message received will be

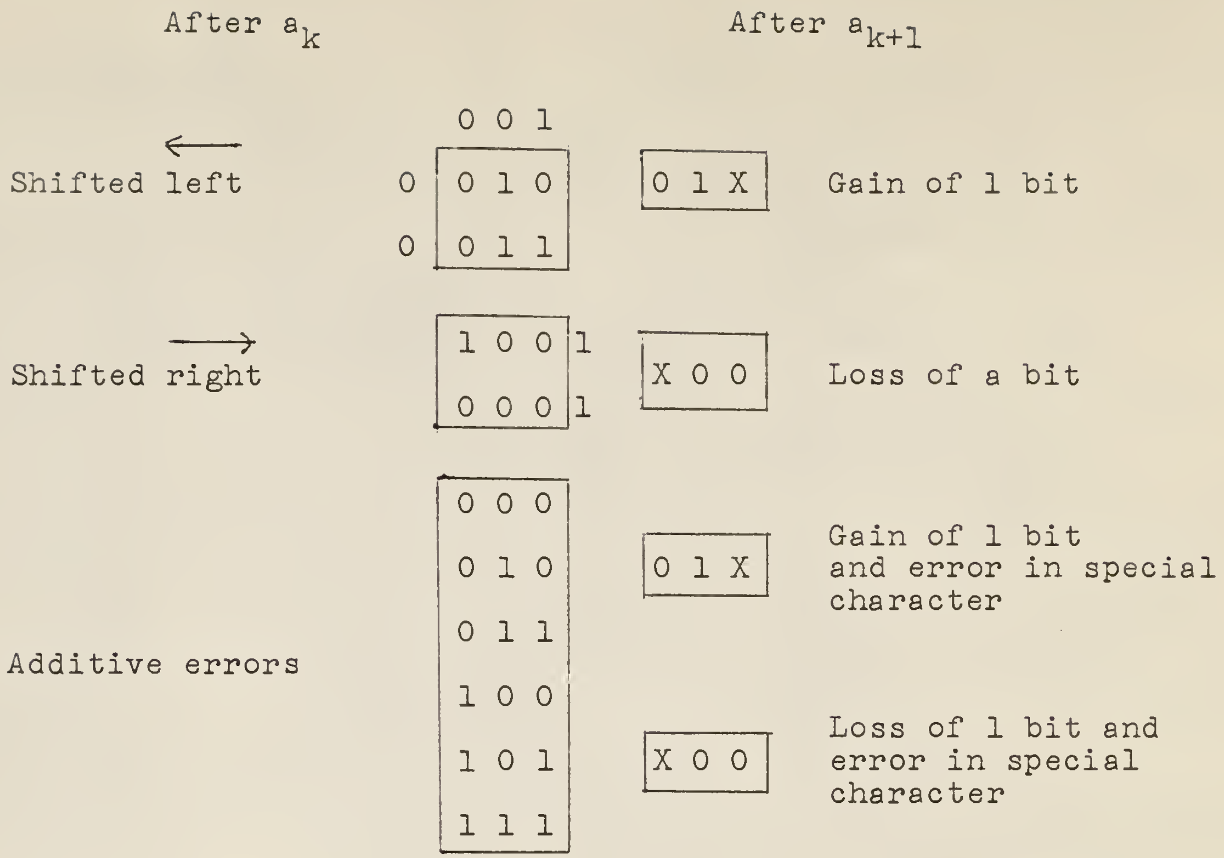
$$1\ 1\ 1\ 1\ \underbrace{1\ 0\ 0}\ 1\ 1\ 1\ 0\ 0\ \underbrace{1\ 0\ 0}\ 1\ 1\ 1\ 0\ 1$$

The rule governing the place to insert the bit is also formulated in accordance with the theorem on the size of burst. In this case the only appropriate place to insert a bit is between a_2 and a_3. If the new bit which would be in position a_3 is in error, the burst still cannot exceed a length of 3, and hence can be corrected. It can be generalized that a bit be inserted between $a_{(J/2-1/2)}$ and $a_{(J/2+1/2)}$ for all odd J. Similarly for even J, the bit will have to be inserted between $a_{(J/2-1)}$ and $a_{J/2}$.

The rules for dependent decoding are essentially the same except as stated above. Additive errors can be corrected and also the information contained in two consecutive special characters is used to determine the correction procedure.

Summary of the Above Discussion (Sellers, 1962)

Special character received	Diagnosis	Correction procedure
001	No error or error in special character	Do nothing this time; (if there is an error, it will be caught by next special character).
010 011	Gain of a bit	Take out received bit. $a_{(J/2+1)}$ for J even $a(\frac{J+1}{2})$ for J odd
100 101	Loss of a bit	Insert a bit between $a_{(J/2-1)}$ and $a_{J/2}$ for J even. $a_{(J/2-1/2)}$ and $a_{(J/2+1/2)}$ for J odd.
101	Gain or loss in special character	Do nothing this time. It will be caught by the next special character.
000 110 111	Additive errors	Impossible



The correction procedure is essentially the same as for independent decoding.

Summary of Above Discussion (Sellers, 1962)

Special character		Diagnosis	Correction procedure
After a_k	After a_{k+1}		
001	Any error	An error occurred between a_k and a_{k+1}	Wait for next special character
01X	01X	Gain of a bit	Take out received bit. $a_{k-(J/2+1)}$ for J even $a_{k-(J/2+1/2)}$ for J odd
000 100 101 111 100	01X	Error in special character and bit gained	Take out a bit in special character after a_k
X00	X00	A bit lost	Add a bit between $a_{k-(J/2-1)}$ and $a_{k-J/2}$ for even. $a_{k-(J/2-1/2)}$ and $a_{k-(J/2-1/2)}$ for J odd.
011 010 100 101 111	X00	Error in special character and bit lost	Insert a bit in special character after a_k

K = J 2 J ......

X = 0 or 1

The Special Character

The length of a special character for multiple bit loss and gain should be greater than or equal to 2b + 1, where b is the burst of length to be corrected, and for multiple bit loss or gain it should be greater than or equal to b + 1.

Take a synchronizing character of length 2b, say 0001. It can be written as two adjacent characters of length b each. Denote the left character as L' and right character as L", i.e.,

$$\overbrace{0\ 0}^{L'}\ \ \overbrace{0\ 1}^{L''}\ .$$

Let the message transmitted be

$$0\ 1\ \overbrace{0\ 0}^{L'}\ \overbrace{0\ 1}^{L''}\ 1\ 1\ 0\ 1\ 1$$

Note that the information following the special character is L". If a loss of 2 bits occurs, the message received would be

$$\overbrace{0\ 1}^{L''}\ \overbrace{0\ 0}^{L'}\ 0\ 1\ 0\ 1\ 1$$

showing that L"L' has replaced L'L". Similarly, if the loss of 2 bits occurs and the information to the right of L" is L', L'L" is received in the position of special character, and hence the error cannot be located.

For the case of bit loss or gain, if the special character was of length 4 and the information following it was a duplicate of special character, say

$$0\ 0\ 0\ 1\ \overbrace{0\ 0}^{L'}\ \overbrace{0\ 1}^{L''}\ 1\ 1\ 0\ 1\ 1$$

and a loss of 4 bits occurs, the message received will be

$$\overbrace{0\ 0}^{L'}\ \overbrace{0\ 1}^{L''}\ 0\ 0\ 0\ 1\ 1\ .$$

In this case the synchronizing character seems to be undisturbed whereas the loss of 4 bits has occurred. Therefore the synchronizing character is of length greater than or equal to 2b + 1 for bit loss and gain of 1 bit, and greater than or equal to b + 1 for bit loss or gain of 1 bit.

Error Correcting Codes for Use with Bit Loss and Gain Code

The choice of error correcting code would depend on whether the channel introduces other errors besides bit loss and gain or not, such as additive errors. A burst error correcting code is used, in case the transmission channel introduces even bursts. In case of the former, the special character locates the approximate position of error and the only requirement is to send in enough redundancy for determining the error pattern. Bits of redundancy are sufficient for the correction of errors caused by length correction. A parity check word called longitudinal redundancy check (LRC) is formed and is used for error detection. When the information is received, length correction is applied and its location in the block is noted. An LRC is calculated on the information and compared with the one received with information. A binary word is formed having '1' where received and calculated LRC differ and '0' where they agree. This is the error pattern.

The number of redundant bits on a message with k information digits is

$$k/j(2L + 1) + j \tag{32a}$$

where (2L + 1) is the length of the synchronizing character. In this equation (2L + 1) will always be odd, i.e., 1, 3, 5, 7,

..., and since the expression $k/j(2L + 1)$ cannot be a fraction of a bit, this will be equal to 1 bit which means there will be $(j + 1)$ bits of redundancy. The redundant bit in addition to j bits of redundancy required to detect the error pattern is the parity check on the information digits.

In order to obtain the optimum value of j or minimum redundancy, Eq.32a is differentiated and equated to zero as below..

$$d/dj\left[k/j(2L + 1) + j\right] = 0 \tag{33}$$

$$-k/j^2(2L + 1) + 1 = 0$$

$$j^2 = k(2L + 1), \text{ or } j = \sqrt{k(2L + 1)} \tag{34}$$

This gives the minimum amount of redundancy. For instance, 100 information digits are to be coded into a code capable of correcting bursts of up to 4 bits.

$$j = \sqrt{k(2L + 1)} = \sqrt{100(2 \times 4 + 1)} = \sqrt{900}$$

$$= 30 \text{ bits}$$

This code will not meet the objective if the additive errors are not in the vicinity of bit loss and gain. If the additive errors occur in the vicinity of bit loss and gain, then a burst error correcting code must be used such as Fire code and Hagelberger code. The Fire code is a cyclic burst error correcting code while Hagelberger code is a recurrent error correcting code. Fire codes have the disadvantage that these require a k-stage buffer storage at the receiver to hold the message to be corrected. With the Hagelberger code this problem is eliminated but complexity of electronic hardware required is directly proportional to the length of bursts to be corrected.

In order to correct a burst of b or fewer bits in a message with k-information digits, Fire code has (2b - 1) + log(n + 1) bits of redundancy where n is equal to k plus number of bits of redundancy; i.e.,

$$n = k + (2b - 1) + \log(n + 1) \qquad (35)$$

A good approximation for the value of j may be obtained by minimizing the numbers of redundancy bits, including special characters. Assume that it is desired to correct all bits of length j - 1 or less.

$$0 = d/dj \left[n/j(2L + 1) + 2(j - 1) - 1 + \log(n + 1) \right] \qquad (36)$$

Substituting $n = k + (2b - 1) + \log(n + 1)$

and $b = (j - 1)$

$$0 = \frac{d}{dj}\left[\frac{k + 2(j-1) - 1 + \log(n+1)}{j} \cdot (2L+1) + 2(j-1) - 1 + \log(n+1) \right]$$

$$= \left[-k/j^2 + 2/j^2 + 1/j^2 - \log(n+1)/j^2 \right] (2L+1) + 2 \qquad (37)$$

$$= \frac{-\left[k - 3 + \log(n + 1) \right] (2L + 1)}{j^2} + 2 = 0$$

$$\text{or} \quad j = \sqrt{\frac{\left[(k - 3) + \log(n + 1) \right] \left[2L + 1 \right]}{2}} \qquad (38)$$

This is the optimum value of j or minimum redundancy.

APPENDIX

Some of the pertinent definitions and rules of algebra of rings and fields (Peterson, 1961) as used in the study of Error Correcting Codes are reviewed.

A group is a system with one operation and its inverse, such as addition and subtraction, or multiplication and division. A group obeys the laws of closure, associativity, and has an identity element as well as inverse of each element. A group which satisfies Commutative law is called an Abelian group. A ring has two operations with the inverse for the first operation only. A ring is an Abelian group under addition and the product of its two elements is defined. It obeys Associative and Distributive laws. A ring is called Commutative if its multiplication operation is Commutative, i.e., for any two elements "a" and "b", ab = ba. A field has two operations both with inverses and is a Commutative ring with a unit element in which every nonzero element has a multiplication inverse.

A set "V" of elements is a vector space over a field "F" if it satisfies the Distributive and Associative laws. It is an Abelian group under addition and the product of the vector "V" and any field element is defined. A set of elements is called a linear associative algebra A, it is a vector space over field F and for any two elements u and v of A, product uv is defined in A. It also satisfies the Associative law and Bilinear law, for any three elements u, v, and w of Z (uv)w = u(vw) (Associative law). If c and d are scalars in F and u, v, and w are vectors in A, then u(cv + dw) = (cuv + duw) and (cv + dw)u = (cvu + dwu) (Bilinear law). Row space of an n x m matrix (aij) is the set of all linear combinations of row vectors and the dimension of the row space is called the row rank. Similarly, column space of a matrix is the linear combination of column vectors of a matrix and is called the column rank.

<u>Example</u> $\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$ Row rank = 3

Column rank = 5

Elementary row operations consist of interchange of any two rows, multiplication of any row by a nonzero field element, and addition of any multiple of any row to another.

<u>Polynomials</u>. A polynomial $f(x) = f_0 + f_1x + \ldots + f_nx^n$ is of degree n. It is called <u>monic</u> if the coefficient of highest power of x is 1. A polynomial p(x) of degree n which is not divisable by any polynomial of degree less than n but greater than zero is called <u>Irreducible</u>. The greatest common divisor of two polynomials is their common polynomial of greatest degree. Two polynomials are said to be <u>relatively prime</u> if their greatest common divisor is 1.

<u>Galois Field</u>. If p(x) is an irreducible polynomial with coefficient in a field F, then the algebra of polynomials over F modulo p(x) is a field. The original field is called a ground field. The field polynomials over GF(p) modulo an irreducible polynomial of degree m is called the Galois Field of p^melements and denoted as GF(p^m). For any number $q = p^m$ that is a power of a prime number, there is a field GF(q) which as q elements. The polynomial has as roots all the (q - 1) nonzero elements of

GF(q) and the polynomial $(x^m - 1)$is divisible by $(x^n - 1)$ if and only if m is divisible by n. In GF(q) there is a primitive element, i.e., an element of order (q - 1). Even a nonzero element can be expressed as a power of α, that is multiplicative group of GF(q) is cyclic. A cyclic group consists of all the powers of one of its elements. For instance, $GF(2^4)$ and its representation are illustrated in Table 11.

This is a cyclic operation repeated after every three elements and note that addition is modulo 2 here.

$$\begin{aligned}
\alpha^3 &= \alpha^3\\
\alpha^4 &= \alpha^0 + \alpha^1\\
\alpha^5 &= \alpha^1 + \alpha^2\\
\alpha^6 &= \alpha^2 + \alpha^3\\
\alpha^7 &= \alpha^3 + \alpha^4 = 1 + \alpha + \alpha^3\\
\alpha^8 &= \alpha^4 + \alpha^5 = 1 + \underbrace{\alpha + \alpha}_{\substack{11\\0}} + \alpha^2\\
&= 1 + \alpha^2 \text{ and so on}
\end{aligned}$$

Every polynomial p(x) of degree n irreducible over GF(q) is a factor of $(xq^m - m)$ and also all the roots of an irreducible polynomial are of the same order.

Polynomial Representation of Binary Information

Consider a code n long in which k are information digits and remaining n - k are parity checks. The message corresponding to the polynomial $(1 + x + x^3 + x^5)$ is 110101. These polynomials obey all polynomial laws except addition which is

Table 11

Elements of $GF(2^4)$	: Representation
$\alpha_0 = 1$	(1000)
$\alpha^1 = \alpha$	(0100)
$\alpha^2 = \alpha^2$	(0010)
$\alpha^3 = \alpha^3$	(0001)
$\alpha^4 = 1 + \alpha$	(1100)
$\alpha^5 = \alpha + \alpha^2$	(0110)
$\alpha^6 = \alpha^2 + \alpha^3$	(0011)
$\alpha^7 = 1 + \alpha + \alpha^3$	(1101)
$\alpha^8 = 1 + \alpha^2$	(1010)
$\alpha^9 = \alpha + \alpha^3$	(0101)
$\alpha^{10} = 1 + \alpha + \alpha^2$	(1110)
$\alpha^{11} = \alpha + \alpha^2 + \alpha^3$	(0111)
$\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$	(1111)
$\alpha^{13} = 1 + \alpha^2 + \alpha^3$	(1011)
$\alpha^{14} = 1 + \alpha^3$	(1001)
$\alpha^{15} = 1$	(1000)

modulo 2, that is

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

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

Addition, multiplication, and division of these polynomials are given below.

Addition

$$\begin{array}{ccccccccc}
1 & + & x & & & + & x^3 & + & x^4 \\
 & & x & + & x^2 & & & + & x^4 \\
1 & + & x & + & x^2 & & & + & x^4 \\ \hline
 & & x & & & + & x^3 & + & x^4
\end{array}$$

Multiplication

$$\begin{array}{ccccccccccc}
1 & + & x & & & + & x^3 & + & x^4 & & \\
1 & + & x & & & & & & & & \\ \hline
1 & + & x & & & + & x^3 & + & x^4 & & \\
 & + & x & + & x^2 & & & + & x^4 & + & x^5 \\ \hline
1 & & & + & x^2 & + & x^3 & & & + & x^5
\end{array}$$

Division

$$\begin{array}{rllllll}
 & 1\cdot x^3 & & & & & \\ \cline{2-7}
\multicolumn{1}{r|}{1 \cdot x^2 + 0 \cdot x + 1} & 1\cdot x^5 & + 1\cdot x^4 & + 1\cdot x^3 & + 0\cdot x^2 & + 1\cdot x & + 0 \\
 & 1\cdot x^5 & + 0\cdot x^4 & + 1\cdot x^3 & & & \\ \cline{2-7}
 & & 1\cdot x^4 & + 0\cdot x^3 & + 0\cdot x^2 & + 1\cdot x & + 0 \\
 & & 1\cdot x^4 & + 0\cdot x^3 & + 1\cdot x^2 & & \\ \cline{3-7}
 & & & 0\cdot x^3 & + 1\cdot x^2 & + 1\cdot x & + 0 \\
 & & & 0\cdot x^3 & + 1\cdot x^2 & + 0\cdot x & \\ \cline{4-7}
 & & & & & 1\cdot x & + 1
\end{array}$$

It is interesting to note that if only the coefficient of the above polynomial is divided, the result is the binary representation, as follows.

$$
\begin{array}{ccccccccc}
 & & & 1 & 1 & 0 & & & \\ \cline{4-9}
1 & 0 & \multicolumn{1}{c|}{1} & 1 & 1 & 1 & 0 & 1 & 0 \\
 & & & 1 & 0 & 1 & & & \\ \cline{4-9}
 & & & & 1 & 0 & 0 & & \\
 & & & & 1 & 0 & 1 & & \\ \cline{5-9}
 & & & & & 0 & 1 & 1 & 0 \\
 & & & & & & 1 & 0 & 1 \\ \cline{6-9}
 & & & & & & & 1 & 1
\end{array}
$$

In modulo 2 operations, $-1 = +1$ and addition and subtraction are equivalent.

BIBLIOGRAPHY

1. Abramson, N. M.
   A class of systematic codes for non-independent errors.
   IRE Trans. PG IT-5. pp. 150-157. 1959.

2. Abramson, N.
   A note on single error correcting binary codes.
   IRE Trans. PG IT-6. pp. 502-503. 1960.

3. Abramson, N.
   Error correcting codes from linear sequential networks.
   Presented at the Fourth London Symposium on Information
   Theory. August, 1960.

4. Birkhoff, G. and S. MacLane.
   A survey of modern algebra. New York: Macmillan, 1941.

5. Bose, R. C. and R. R. Knebler, Jr.
   On the construction of a class of error correcting binary
   signalling codes. Technical Report, University of North
   Carolina, Chapel Hill, North Carolina. May, 1958.

6. Bose, R. C. and D. K. Ray Chaudhuri.
   On a class of error correcting binary group codes.
   Inf. and Control (3), 68-79. 1960.

7. Bose, R. C. and D. K. Ray Chaudhuri.
   Further results on error correcting binary group codes.
   Information and Control (3), 279-290. 1960.

8. Brown, D. T. and W. W. Peterson.
   Cyclic codes for error detection. IRE Proc. (49), 228-
   235. January, 1961.

9. Elspas, B.
   The theory of autonomous linear sequential networks.
   IRE Trans. CT-6, pp. 45-60. 1959.

10. Elspas, B.
    A note on p-nary adjacent error correcting codes.
    IRE Trans. PG IT-6. pp. 13-15. 1960.

11. Fire, P.
    A class of multiple error correcting binary codes for
    non-independent errors. Sylvania Report, RSL-E-2.
    Sylvania Reconnaissance Systems Laboratory, Mountain
    View, California. 1959.

12. Golay, M. J. E.
    Binary coding. IRE Trans. PG IT-4. pp. 23-28. 1954.

13. Hagelberger, D. W.
Recurrent codes: easily mechanized burst correcting binary codes. Bell System Tech. Journal (38). pp. 969-984. 1959.

14. Hamming, R. W.
Error detecting and error correcting codes. Bell System Tech. Journal (29). pp. 147-160. 1950.

15. Hancock, J. C. and J. L. Holsinger.
Some useful binary coding techniques. Purdue University, School of Electrical Engg., TR-EE-62-1. January, 1962.

16. McCluskey, E. J., Jr.
Error correcting codes - a linear programming approach. Bell System Tech. Journal (30). pp. 1485-1512. 1959.

17. Table of irreducible polynomials over GFC2 through degree 19. NSA, Washington, D.C. 1957.

18. Meggitt, J. E.
Error correcting codes for correcting burst of errors. IBM Journal, Research and Development (4). pp. 329-334. 1960.

19. Melas, C. M.
A new group of codes for correction of dependent errors in data transmission. IBM Journal, Research and Development (4). pp. 58-65. 1960.

20. Metzner, John J.
Burst error correction for randomly chosen binary group codes. IEEE Trans. IT-9. pp. 281-285. October, 1963.

21. Melas, C. M.
A cyclic code for double error correction. IBM Journal, Research and Development (4). pp. 364-366. 1960.

22. Peterson, W. W.
Error correcting code. M.I.T. Press. 1961.

23. Prange, E.
Cyclic error correcting codes in two symbols. AFCRC-TN-57-103. Cambridge, Mass.: Air Force Cambridge Research Center. September, 1957.

24. Prange, E.
Some cyclic error correcting codes with simple decoding algorithms. AFCRC-TN-58-156. Bedford, Mass.: Air Force Cambridge Research Center. April, 1958.

25. Reed, I. S. and G. Solomon
Polynomial codes over certain finite fields. J. Soc. Indust. Appl. Math. (8). pp. 300-304. 1960.

26. Reed, I. S.
A class of multiple error correcting codes and decoding scheme. IRE Trans. PG IT-4. pp. 38-49. 1954.

27. Reiger, S. H.
Codes for the correction of clustered errors. IRE Trans. PG IT-6. pp. 16-21. 1960.

28. Sacks, G. E.
Multiple error correction by means of parity checks. IRE Trans. PG IT-4. pp. 145-147. 1958.

29. Sellers, F. F., Jr.
Bit loss and gain correction code. IRE Trans. PG IT-8. pp. 35-38. 1962.

30. Shapiro, H. S. and D. L. Slotnick.
On the mathematical theory of error correcting codes. IBM Journal of Research and Development (3). pp. 25-34. 1959.

31. Slepian, D.
A class of binary signalling alphabets. Bell System Tech. Journal (35). pp. 203-234. 1956.

32. Slepian, D.
A note on two binary signalling alphabets. IRE Trans. PG IT-2. pp. 84-86. 1956.

33. Slepian, D.
Some further theory of group codes. Bell System Tech. Journal (39). pp. 1219-1252. 1960.

A STUDY OF BINARY CODES

of

VIRENDRA K. MANAKTALA

B. E. (Electrical),
University of Delhi, India, 1961

AN ABSTRACT OF
A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1965

In a communication system which usually consists of a source, encoder, channel, decoder, and a receiver, an encoder is used to improve the efficiency of the channel by increasing the immunity of transmitted message, whereas a decoder decodes the received signal to recover the original message. Communication channels are either noiseless or noisy, such as binary symmetric, binary erasure, and cascaded channel. A noiseless channel has one and only one nonzero element in each column of its channel matrix. A deterministic channel has one and only one nonzero element in each row of its channel matrix. This report covers binary codes for memoryless discrete sources using noiseless and noisy channels.

Binary encoding procedures for discrete memoryless sources and the noiseless channels by Shannon-Fano, Shannon, Gilbert-Moore, and Huffman are not devised to detect or correct any errors in a received message but instead to optimize the coding efficiency. Shannon-Fano codes are 100 per cent efficient when the probability of occurrence of each message X_k is of the form

$$P(X_k) = 2^{-n_k}$$

where n_k is any positive integer. Shannon's binary codes exist for such sources if the inequality

$$\sum_{k-1}^{n} 2^{-n_k} \leq 1$$

is satisfied, whereas Gilbert-Moore encoding exists for source alphabets which satisfy the following inequality:

$$2^{1-n_k} \leq P(X_k) \leq 2^{2-n_k}, \; k = 1, 2, \ldots, N$$

Huffman's minimum redundancy code is the most efficient code for a specified source, where minimum redundancy implies an optimum code defined as a code with minimum average word length for a given source probability matrix.

Slepian (1956) generalized the error detecting and correcting codes introduced by Hamming (1950) and Reed-Muller. These codes now form a subclass of a larger class of codes called Group Codes. Bose-Chaudhuri codes are the general form of all these codes and their existence theorem states that, "for any m and t, there exists a Bose-Chaudhuri code of length 2^{m-1} which corrects all combinations of t or fewer errors and has no more than mt parity check symbols" (Peterson, 1961). Reed-Muller codes cover a wide range of rate and minimum distance. Fire codes can be used for correcting a single burst of error while Reed-Solomon code is capable of correcting more than one burst of errors. The average length of the latter is shorter than that of the former for correcting the same length of burst.

Sellers (1962) introduced a block code that can correct an error caused by gain or loss of a bit within the block. This code is constructed by inserting a special character into a burst error correcting code at periodic intervals. This special character locates the approximate position of the bit loss or gain. From that location, a bit is inserted or removed from the block depending on whether a loss or a gain has occurred. The error correcting code then corrects the erroneous bits between where the error occurred and where the correction took place. This code can be generalized to correct the loss or gain of a burst of bits. Since no bounds have been derived, the efficiency of this method is not known as compared to other possible codes.