# A CLASS OF BINARY SINGLE ERROR-CORRECTING CODES

by

Richard Provost

Submitted to the Department of Electrical Engineering in partial fulfilment of the requirements for the degree

of

Master of Applied Science

Department of Electrical Engineering
Faculty of Pure and Applied Science
University of Ottawa
Ottawa, Ontario

August, 1971.

## ABSTRACT

This thesis discusses some useful results on one-step majority-logic decodable binary codes and especially a class of binary single error-correcting codes which are one-step majority-logic decodable. This class is as large as a known class of self-orthogonal quasicyclic codes for single error correction.

ACKNOWLEDGEMENT

The author would like to thank Professor S.G.S. Shiva for his guidance, patience, and encouragement throughout this research.

Thanks are also due to the people of the Electrical Engineering Department of the University of Ottawa and the National Research Council for financial assistance.

... (iii)

# TABLE OF CONTENTS

# INTRODUCTION

In a communication system, we are dealing with the problem of transmitting information from one place to another. For example, one could require transmission of information between computers, between business machines and computers, and between spacecraft and ground stations.

Almost without exception, communication between machines employs electrical signals, and the information to be conveyed is digital. The discipline of data communications deals with the analysis and design of systems for transmitting this digital information. Although digital communication has been used in various rudimentary forms for centuries, i.e. the repetition code , the science of data communications is relatively new and developing rapidly.

This rapid development is due to two related demands. The amount of data to be transmitted is increasing almost exponentially with time, and because of the complex ways in which the data are used, higher transmission accuracies are necessary. Neither of these demands can be met solely by building more and better communication facilities. Rather, it is necessary to employ existing facilities with as high a data rate and an accuracy as possible. Thus, the emphasis in the discipline of data communications is on the efficient use of transmission facilities.

From a technical point of view, we can regard the data-communications system as consisting of three basic blocks: the

transmitter, the channel, and the receiver. (See Fig. 0.1)


FIG. 0.1: BLOCK DIAGRAM OF A DATA COMMUNICATIONS SYSTEM.

The transmitter has the task of assigning an electrical waveform to each possible sequence of digits received as input from the data source. The electrical waveform is then passed through the channel, which may typically be a telephone connection, a satellite link, a microwave system, or a radio link. In passage through the channel, the transmitted waveform is invariably corrupted by noise. Because of this noise the received waveform does not correspond exactly to any of the possible transmitted waveforms. Nevertheless, the receiver must make a decision as to which of the data sequences is most likely to have given rise to the particular received waveform.

To minimize the probability of making a wrong decision at the receiver, the waveforms selected for transmission must be ones which are not likely to be confused by the receiver. One obvious way to accomplish this is to choose signals which have a great deal more power than the average noise waveform. However, the amount of signal power which can be employed is always limited (sometimes by the power available at the transmitter, sometimes by restrictions imposed by the channel). Thus, it is necessary to seek a more subtle means of

... 3

avoiding transmission errors within the constraint of allowed signal power.

One solution to the problem lies in the law of large numbers. This law states that while the outcome of a single chance event may fluctuate widely, the overall result of many repetitions of this chance event is subject to very accurate prediction. In data communications, this law can be used to advantage when the duration of the input signals is made very long. Certain statistical properties of noise, such as its power, then become much more predictable than if a short signal had been used to convey the information. Unfortunately, this reduced noise effect comes at a great cost in the complexity of the transmission and receiver since for a given rate of data transmission, the number of possible messages grows exponentially with the length of the signal.

Because of this complexity with signal duration the assignment of long message waveforms in the direct manner mentioned above have not been attempted in practice. Instead, the waveforms are selected at frequent intervals so that only a small number of basic signals are needed (often two). The process of matching the input digital sequence into a train of short analog waveforms is known as modulation. At the receiver, the basic waveforms are interpreted individually so that the output of the detector is a sequence of digits representing best guesses of the transmitted data. This classic method of transmitting data foregoes any possibility of obtaining virtually error-free transmission through the use of long, basic signalling intervals.

... 4

With the above general type of modulation, we can view the communication system of Fig. 0.1 in its entirety as a strictly digital channel which we call the coding channel. In the binary case, which is the only case discussed in this thesis, the channel accepts successive 0's and 1's at its input and usually reproduces them at the output. Sometimes, however, because of noise or other channel impairements, the output digits do not agree with the input digits. The digital error rate is the average number of disagreements between the output and input digits.

Whereas on the modulation channel each possible message was associated with a signal waveform, on the coding channel each message (or block of digits from a message sequence) must be associated with a sequence of digits. Just as our goal in choosing signals was to minimize the probability that waveforms would be confused, so now do we desire to associate with messages digit sequences which are as different as possible from one another. This idea of "as different as possible" is a very fundamental idea in coding theory. For example, suppose that one of the $2^k$ possible equally likely messages, i.e. a sequence of k digits, is to be transmitted over a binary channel. The binary digit sequence associated with each message must be at least k digits long. If it is exactly this long, every possible received k digit sequence corresponds to a sequence associated with some message. Therefore, any error introduced into a sequence by the channel will cause the received message to correspond to a different message from the transmitted sequence. This vulnerability to error can be reduced substantially by alloting a large number of digit intervals, say n, to

... 5

the transmission of each of the $2^k$ messages. These extra (n-k) digits are called the parity check digits. Since usually a small fraction, $(2^k)/(2^n)$, of the possible transmitted sequences correspond to messages, it is very possible to choose these sequences to be quite unlike one another. This makes the communications system considerably less vulnerable to channel errors.

The process of associating an n-bit sequence with a k-bit message is called encoding. The inverse operation, decoding, is performed at the receiver and consists of associating a k-bit message with each received n-bit block. An erroneous decoding is committed if the decoder fails to reproduce the actual transmitted code word. The probability of an erroneous decoding depends on the code used, the channel characteristics, and the decoding strategy employed at the decoder. If all the code words have equal likelihood of being transmitted, the best decoding scheme is as follows. Upon receiving the sequence, the decoder computes the conditional probability P(R/C), where R is the received sequence at channel output and C is the transmitted code word, for all possible $2^k$ transmitted code words. The code word $C_t$ is identified as the transmitted word if the conditional probability $P(R/C_t)$ is the largest. This decoding scheme is known as maximum likelihood decoding. The general data communications system of Fig. 0.1 now has the form of Fig. 0.2

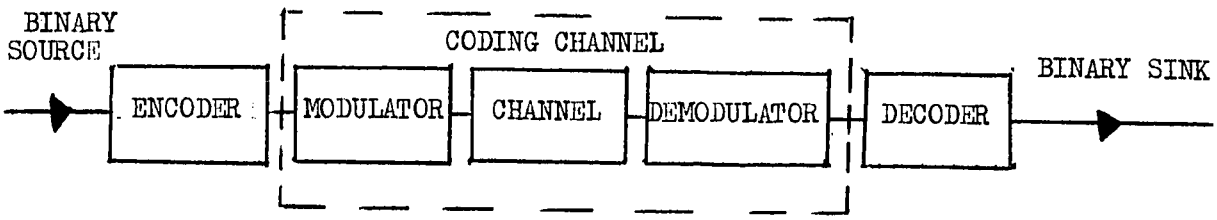


Fig. 0.2: Augmented BLOCK DIAGRAM of a DIGITAL COMMUNICATIONS CHANNEL.

... 6

Since n is greater than k, coding results in a net reduction in data rate. The ratio R=k/n is known as the efficiency of the code. This trading of data rate for error rate is a property common to all coding systems. But by using very long sequences, say up to thousands or tens of thousands of bits in length, it is often possible to attain a very low error rate at the expense of only a slight reduction in data rate. This most important and most striking result on transmission of information over a noisy channel is due to Shannon. [1,2] Shannon's "Coding Theory" says that every channel has a definite capacity C, and that for any rate R less than C, there exists codes of rate R which, with maximum likelihood decoding, have an arbitrarily small probability of erroneous decoding P(E). More specifically, for any given rate $R<C$ and length n, there exists a block code, which uses sequences of length n, such that the probability of erroneous decoding is

$$P(E) \leq e^{-n\,E(R)}$$

where E(R) is a positive function of R for $R<C$ and is specified by the channel transition probabilities. Therefore, the probability of decoding error can be made as small as we desire by increasing the code length n, and keeping $R<C$. Shannon's theorem only shows the existence of codes which give arbitrarily small probability of decoding error, but does not indicate how these codes can be constructed. We are still left with the problem of how to construct these good codes of large n promised by Shannon.

At this point, one may ask why we have introduced the idea of coding since the problem of associating one of a large set of signals with the message at the receiver and the inverse operation at the

receiver remains. Have we not just substituted a hopelessly complex analog machine for an equally costly digital machine? The answer is no. First of all the encoding process now typically consists of associating with a k-bit message sequence a somewhat longer n-bit sequence. Because of the digital nature of this association, it can be performed according to a very simple set of rules. Therefore, instead of having to search through the entire set of signals to find the one associated with a particular message, the encoder can calculate the sequence to be transmitted from the message sequence. Similarly, decoding is also simpler in the digital domain, although it is in general more complex than encoding. A second advantage of digital coding is that it is simpler and more economical to process and store than analog signals. Therefore, the coding theorist is mainly interested in finding codes which have the simplest encoding and decoding rules.

As is well known majority-logic decoding (MLD), the meaning of which will be made clear later in this thesis, is among the simplest to implement among available decoding techniques. Hence, codes which are so decodable are of interest.

Among known majority-logic-decodable codes, one class is due to Townsend and Weldon.[3] These codes are self-orthogonal and quasicyclic. Hereafter these codes will be referred to as TW codes. In the present thesis, we give some useful results on one step majority-logic-decodable binary block codes and a class of single error correcting one step majority-logic-decodable binary block codes V. This class is as large as the class of TW codes for single error correction. In fact, the values of n (length) and k(number of information or message bits) are identical with those of TW codes. However V is generated by a single

polynomial like a cyclic code and is, therefore, somewhat simpler to implement than TW codes.

It is possible to lengthen V to make it majority-logic-decodable in more than one step. We do not have any general rules in this regard. However, we give three examples for the two step case.

In summary, the reason for separating the general communication problem into two parts, modulation and coding, is the following. In order to use communication channels efficiently, long signals must be employed. Since the number of signals grows exponentially with their length, the transmitter and receiver must be capable of handling an erroneous number of signals. The only practical way these signals can be processed and stored is by quantizing the allowed signal values and using digital circuitry. Fortunately, this quantization does not necessarily result in much loss in capability of the system. Furthermore, simple digital coding rules are known which circumvent the problem of exponential growth in complexity with message length. Majority-logic-decoding is the simplest to implement among available decoding techniques and is the main concern of this thesis.

# CHAPTER 1

## LINEAR BLOCK CODES. [4,5,6]

### Block Codes.

In the "Introduction", we saw that each block of k successive information digits would be transformed at the encoder according to certain rules to a longer block of n (n>k) binary digits (binary n-tuple) which we call a code word. Since each message block consists of k binary digits, there are $2^k$ possible distinct information blocks. Therefore, corresponding to the $2^k$ possible messages, there are $2^k$ possible code words at the output of the encoder. This set of $2^k$ code words is called a block code. A code word is often called a code vector because it is an n-tuple from the vector space of all n-tuples. In the following, we shall consider codes with the structure that the $2^k$ code words of each code form a k-dimensional subspace of all n-tuples. A set of $2^k$ code words is called a linear block code if and only if it is a subspace of the vector space of all n-tuples.

### Generator Matrix.

For a subspace, it is possible to find a set of linearly independent n-tuples, say k of them, $A_1$, $A_2$, ..., $A_k$, such that each n-tuple of the subspace is a linear combination of the $A_i$'s in the following form:

$$B=m_1 \cdot A_1 + m_2 \cdot A_2 + \ldots + m_k \cdot A_k \qquad (1.1)$$

where $m_i$ belongs to Galois Field of two elements "0" and "1", i.e. GF(2), for i=1 to k. This is a k-dimensional subspace and consists of $2^k$ n-tuples. Then, a linear block code of $2^k$ code words can always

be described by a set of k linearly independent code vectors. Let us arrange these k independent code words as rows of a k x n matrix.

$$G = \begin{bmatrix} A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_k \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots\cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots\cdots & a_{2,n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ a_{k,1} & a_{k,2} & \cdots\cdots & a_{k,n} \end{bmatrix} \qquad (1.2)$$

where $A_i = (a_{i,1}, a_{i,2}, \cdots\cdots, a_{i,n})$, i=1 to k and $a_{i,j}$ belongs to GF(2) for j=1 to n. Let $M = (m_1, m_2, \cdots\cdots, m_k)$ be an information block. Then, the corresponding code word can be given as

$$B = M \cdot G = (m_1, m_2, \cdots\cdots, m_k) \begin{bmatrix} A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_k \end{bmatrix} \qquad (1.3)$$

$$= m_1 \cdot A_1 + m_2 \cdot A_2 + \cdots\cdots + m_k \cdot A_k.$$

We see that the code word corresponding to M is a linear combination of the rows of G. Therefore, the rows of matrix G generate a linear block code. The matrix G is known as the generator matrix of the linear block code. This linear block code is called an (n,k) code, i.e. a block of k information digits is encoded into a code word of n digits to be transmitted over the noisy channel. The ratio R=k/n is called the code rate or efficiency.

A Systematic Code.

It is possible to encode each message block into a code word in such a way that the first k digits of the code word are

exactly the same as the message block and the last n-k digits are redundant digits which are functions of information digits. A code of this form is called a systematic code. A systematic (n,k) linear block code can be described by a kxn matrix of the following form:

$$G_s = \begin{bmatrix} 1\,0\,0\,\ldots\,0 & p_{1,1} & p_{1,2} & \ldots & p_{1,(n-k)} \\ 0\,1\,0\,\ldots\,0 & p_{2,1} & p_{2,2} & \ldots & p_{2,(n-k)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0\,0\,0\,\ldots\,1 & p_{k,1} & p_{k,2} & \ldots & p_{k,(n-k)} \end{bmatrix} \quad (1.4)$$

$$= [I_k, P]$$

where $p_{i,j}$ belongs to GF(2) for i=1 to k and j=1 to n-k, and $I_k$ is the kxk identity matrix and P is the kx(n-k) matrix of $p_{i,j}$'s. Consider an information block $M = (m_1, m_2, \ldots\ldots, m_k)$. By using $G_s$ of 1.4 the corresponding code word is:

$$A = M \cdot G_s \quad (1.5)$$

By matrix multiplication we find that:

$$a_i = m_i \quad \text{for} \quad i = 1 \text{ to } k \quad (1.6)$$

$$a_{k+j} = p_{1,j} \cdot m_1 + p_{2,j} \cdot m_2 + \ldots\ldots + p_{k,j} \cdot m_k \text{ for } j = 1 \text{ to } n-k \quad (1.7)$$

From these last two equations, we see that the first k digits of the code word are the information digits and that the last n-k digits are linear functions of the information digits. The last n-k digits of A are called the parity-check digits of the code word. The equations 1.7 are called the parity-check equations of the code.

... 12

Parity-Check Matrix.

For each k x n matrix G, there exists an $(n-k) \times n$ matrix H such that the row space of G is orthogonal to H. By orthogonality, we mean that the dot product or inner product of any vector in G with any vector in H is zero. Now if A is a vector in the row space of G, then

$$A \cdot H^T = (0,0, \ldots\ldots, 0) \qquad (1.8)$$

where T stands for transpose. Then A is said to be in the code generated by G if and only if $A \cdot H^T = (0,0, \ldots\ldots, 0)$. The matrix H is called the parity-check matrix of the code.

If the generator matrix of a systematic code is of the form of equation 1.4, then the parity-check matrix of this code is

$$H_s = \left[P^T, I_{n-k}\right] \qquad (1.9)$$

where $P^T$ is the transpose of matrix P and $I_{n-k}$ is the $(n-k) \times (n-k)$ identity matrix. The parity-check equations of 1.7 can also be obtained from $H_s$ of 1.9. Therefore, a linear block code can be uniquely specified either by its generator matrix or by its parity matrix.

Error-Correcting Capability of Linear Codes.

Some of the basic terminologies used to define the error-correcting capability of a linear code are introduced at this point.

The Hamming weight of an n-tuple A, w(A), is defined as the number of non-zero components of A.

i.e. If A = (1 1 0 1 1 0 1), then w(A) = 5.

Let A and B be two n-tuples. Then the Hamming distance between A and B, d(A,B), is defined as the number of components in which

... 13

they differ.

i.e. If A = (1 1 0 1 1 0 1) and B = (0 0 1 1 1 0 1), then d(A,B) = 3.

By the definition of modulo-2 addition, it can be easily seen that

$$d(A,B) = w(A \oplus B) \qquad (1.10)$$

Given a linear code, we can calculate the distances between all possible pairs of code words. The smallest distance is called the minimum distance, $d_{min}$, of the code. In a linear code the sum of two code words is also a code word since a linear code is a vector space. Therefore, the distance between two code words is equal to the weight of a third code word. Thus, the minimum distance of a linear code is equal to the minimum weight of its non-zero code words.

In certain communication channels, transmission errors occur independently, i.e. each transmitted symbol is affected independently by noise. Errors of this nature are called random errors. Codes which are designed to combat independent errors are called random error-correcting codes.

There are also channels (telephone lines, magnetic-tape storage systems, etc...) on which the disturbances introduce errors of unspecified time duration, i.e. errors tend to cluster together in bursts. These kinds of errors are called burst errors. Codes which are designed to correct burst errors are called burst error-correcting codes.

If a code with minimum distance $d_{min}$ such that $2t + 2 \geq d_{min} \geq 2t + 1$ is used for random error correction, the decoder can correct all error patterns of t or fewer errors which may occur in the received word.

... 14

This can be justified as follows. Let A be the transmitted code word, B be the received word, and C be any other code word. Then, the Hamming distances among A,B,C satisfy the following inequality.

$$d(A,B) + d(C,B) \geq d(A,C) \tag{1.11}$$

Suppose an error pattern of $t^1 \leq t$ errors occurs

Then
$$d(A,B) = t^1 \tag{1.12}$$

Since
$$d(A,C) \geq d_{min} \geq 2t + 1 \tag{1.13}$$

then by equation 1.11

$$\begin{aligned} d(C,B) &\geq 2t + 1 - t^1 \\ &\geq t + 1 \\ &> t^1 \end{aligned} \tag{1.14}$$

This inequality says that if an error pattern of t or fewer errors occurs, the received word B is closer to the actual transmitted word A than to any other code word C. Thus, the decoder will make a correct decoding if the decoder selects the code word having the smallest distance to the received word. On the other hand the decoder cannot correct all the error patterns of $\ell$ errors $\ell \geq t + 1$, for there is at least one case where an error pattern of $\ell$ errors results in a received word which is closer to an incorrect word than to the transmitted code word. In this case the decoder will make an incorrect decoding. Thus, we say that the above code has error-correcting capability t. In general, a code with minimum distance, $d_{min}$, has error-correcting capability $t = \left[\frac{d_{min} - 1}{2}\right]$ where $[x]$ denotes the largest integer not greater than x. A code

... 15

of error-correcting capability t is generally called a t-error-correcting code. A linear (n,k) code with t-error-correcting capability is denoted as an (n,k,t) code.

Polynomial Representation.

Suppose that we have a code word

$$A = (a_0, a_1, \ldots\ldots, a_{n-1})$$

belonging to an (n,k) linear code. We can treat the components of the code word as coefficients of a polynomial as follows:

$$A = (a_0, a_1, \ldots\ldots, a_{n-1}) \Longleftrightarrow A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1} \quad (1.15)$$

Thus, each code word corresponds one-to-one to a polynomial of degree n-1 or less. We shall call A(x) the code polynomial of A. Hereafter, we shall use the terms code word and code polynomial interchangeably.

Sometimes it is possible to represent the generator matrix of a code as

$$G = \begin{bmatrix} g(x) \\ x \cdot g(x) \\ \vdots \\ x^{k-1} \cdot g(x) \end{bmatrix} \quad (1.16)$$

where $g(x) = 1 + g_1x + g_2x^2 + \ldots\ldots + g_{n-k-1}x^{n-k-1} + x^{n-k}$ and $g_i$ belongs to GF(2) for i=1 to n-k-1. Then a code word A(x) can be represented as a linear combination of the rows of G in the following way:

$$A(x) = M(x) \cdot g(x)$$

$$= (m_0 + m_1x + \ldots\ldots + m_{k-1}x^{k-1}) \cdot g(x) \quad (1.17)$$

... 16

where $(m_0, m_1, \ldots\ldots, m_{k-1})$ are the k information digits. M(x) is known as the information polynomial of the code.

Linear Cyclic Codes.

Now an (n,k) linear code V is called a cyclic code if it has the following property:

If an n-tuple $A = (a_0, a_1, \ldots\ldots, a_{n-1})$ belongs to V, the n-tuple $A^1 = (a_{n-1}, a_0, a_1, \ldots\ldots, a_{n-2})$, obtained by shifting A cyclically one place to the right, is also a code word of V, i.e.

$x \cdot A(x)$ modulo $1 + x^n = A^1(x)$ for any A(x) belonging to V. In an (n,k) cyclic code one can show that all code words are multiple of a unique generator polynomial of degree n-k and the generator polynomial is a factor of $1 + x^n$, i.e.

$$1 + x^n = g(x) \cdot h(x) \qquad (1.18)$$

Suppose that we have an(n,k) cyclic code. Consider the set of code words whose y($y<k$) leading high-order information digits are zero. There are $2^{k-y}$ such code words. If the y zero information digits are deleted from those code vectors, we obtain a set of $2^{k-y}$ code words of n-y digits. It is easy to see that this set of shortened code words form an (n-y, k-y) linear code. This code is called a shortened cyclic code and is not cyclic. A shortened cyclic code has at least the same error-correcting capability as the code from which it is derived.

In this chapter, we have reviewed some of the basic ideas relevent to linear block codes such as the generator and parity-check matrix, the error-correcting capability of the codes and the class of cyclic codes.

... 17

# CHAPTER 2

## MAJORITY-LOGIC DECODING [7,8,9]

As is well known majority-logic decoding is one of the simplest to implement among available decoding techniques. [4,5,6] Hence codes which are so decodable are of interest.

### One-Step Majority-Logic Decoding.

Consider the (7,3,1) cyclic code V generated by $g(x) = 1 + x^2 + x^3 + x^4$. Let v(x) belong to V. Then

$$H_s = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.1)$$

where $v(x) = (r_0, r_1, \ldots, r_6)$,
so that $v(x) \cdot H^T = (0,0,0,0)$ gives

$$\left.\begin{aligned} r_0 + r_1 + r_3 &= 0 \\ r_1 + r_2 + r_4 &= 0 \\ r_0 + r_1 + r_2 + r_5 &= 0 \\ r_0 + r_2 + r_6 &= 0 \end{aligned}\right\} \qquad (2.2)$$

These equations 2.2 are known as the parity-check sums on the received digits. From them we obtain directly the following three equations:

$$\left.\begin{aligned} r_0 &= r_0 \\ r_0 &= r_1 + r_3 \\ r_0 &= r_2 + r_6 \end{aligned}\right\} \qquad (2.3)$$

These equations 2.3 have only one received digit in common, i.e. $r_0$. Therefore, these equations are said to be orthogonal on $r_0$. Since this code is cyclic, such orthogonal equations can be formed for all $r_i$'s and the code is said to be completely orthogonalizable. If no more than one error has occurred during transmission, i.e. only one digit is erroneous, then each $r_i$ can be estimated correctly by a majority vote. This code is then said to be one-step majority-logic decodable single error-correcting. If, for a completely orthogonalizable code, D equations like those of 2.3 can be formed then at least $t = \left[\frac{D-1}{2}\right]$ errors can be corrected since t errors can affect at most $\frac{D-1}{2}$ estimates of $r_0$ and, therefore, at least $\frac{D+1}{2}$ estimates are bound to be correct. In general, therefore, it is possible to have a one-step majority-logic decodable $t = \left[\frac{D-1}{2}\right]$ error correcting codes.

One-Step Majority-Logic Decodable Codes.

One step majority-logic decoding is most effective for cyclic codes which are completely orthogonalizable. Unfortunately, there exists very few good cyclic codes in this category. In the following two small classes of one-step completely orthogonalizable cyclic codes are presented.

(i) Maximum-Length Codes.

For any integer $m \geq 2$, there exists a maximum

... 19

length code with the following parameters:

$$\left.\begin{aligned} n &= 2^m - 1 \\ k &= m \\ d_{min} &= 2^{m-1} \end{aligned}\right\} \qquad (2.4)$$

The generator polynomial of this code is

$$g(x) = (1+x^n) \;/\; p(x), \qquad (2.5)$$

where $p(x)$ is any primitive polynomial of degree m. By a primitive polynomial of degree m, we mean a polynomial of degree m that gives a complete table with $2^m$ distinct symbols containing "0" and "1". [4,5,6] Maximum-length codes were first shown to be majority-logic decodable by Yale [10] and Zierler [11] independently.

(ii) Difference-Set Codes.

The formulation of difference-set codes is based on the construction of perfect difference-sets. A difference-set D of order S and modulo $m \geq S(S-1)+1$ is defined as a collection of S integers chosen from the set $(0,1,\ldots,m-1)$ such that no two of the $S(S-1)$ ordered differences modulo m are identical. If $m = S(S-1)+1$, then for any non-zero integer $n < m$ there is exactly one pair of elements in the difference set such that their difference is congruent to n modulo m. Such a set is called a perfect difference-set. Singer [12] has shown how to construct such sets when $S = p^q + 1$, where p is prime and q is any positive integer. In the

...20

following, we shall only be concerned with $S = 2^q + 1$.

Let $D = (d_0 = 0, d_1, d_2, \ldots, d_{S-1})$ be a perfect difference-set of order $S = 2^q + 1$. Define the following polynomial:

$$D(x) = 1 + x^{d_1} + x^{d_2} + \ldots + x^{d_{S-1}} \qquad (2.6)$$

Let $n = 2^q(2^q + 1) + 1$ be the block length and $h(x)$ be the greatest common divisor of $D(x)$ and $1 + x^n$, i.e.

$$h(x) = \text{GCD}\ (D(x),\ 1 + x^n) \qquad (2.7)$$

Then a difference-set code of length n is defined as the cyclic code generated by

$$g(x) = (1 + x^n) \ / \ h(x) \qquad (2.8)$$

with the following parameters:

$$\left.\begin{aligned} &n = 2^{2 \cdot q} + 2^q + 1, \\ &n - k = 3^q + 1, \text{ and} \\ &d_{min} = 2^q + 2 \end{aligned}\right\} \qquad (2.9)$$

Difference-set codes were discovered by Rudolph [13,14] and Weldon [15] independently. The formula for the number of information digits, k, was derived by Graham and McWilliams. [16]

(iii) Self-Orthogonal Quasicyclic Codes.

In this section, a definition of self-orthogonal quasicyclic codes is given. For further information the reader is asked to consult reference [3].

...21

A quasicyclic (n,k) code is defined as a linear block code of length $n = m \cdot n_0$ and efficiency $R = k/n$ which possesses the following properties: 1) each $n_0$ symbol section (subblock) of a code word is composed of $k_0 = R \cdot n_0$ information symbols followed by $n_0 - k_0$ parity checks; 2) each cyclic shift of a code word by $n_0$ symbols yields another code word.

If the parity check rules are chosen so that no two information symbols appear together in more than one parity-check equation, then the code is said to be self-orthogonal (see Massey [7] ). If each information symbol in such a code is checked by at least d-1 parity checks, the minimum distance between code words is at least d (see Massey [7] p.6). These codes are also based on the idea of being able to find perfect difference-sets.

L-Step Majority-Logic Decoding.

Consider the (7,4,1) cyclic code V generated by $g(x) = 1 + x + x^3$. The parity-check matrix H is found to be

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad (2.10)$$

... 22

Let $v(x) = r_0 + r_1 x + \ldots\ldots + r_6 x^6$ belong to V. Then

$$v(x) \cdot H^T = (0,0,0,0) \quad \text{gives}$$

$$\left.\begin{aligned} r_0 + r_3 + r_5 + r_6 &= 0 \\ r_1 + r_3 + r_4 + r_5 &= 0 \\ r_2 + r_4 + r_5 + r_6 &= 0 \end{aligned}\right\} \qquad (2.11)$$

From 2.11 we obtain

$$\left.\begin{aligned} r_5 + r_6 &= r_5 + r_6 \\ r_5 + r_6 &= r_0 + r_3 \\ r_5 + r_6 &= r_2 + r_4 \end{aligned}\right\} \qquad (2.12)$$

Also from 2.11 we obtain

$$\left.\begin{aligned} r_4 + r_5 &= r_4 + r_6 \\ r_4 + r_6 &= r_0 + r_1 \\ r_4 + r_6 &= r_2 + r_5 \end{aligned}\right\} \qquad (2.13)$$

Let $S_1 = r_5 + r_6$ (2.14)

and $S_2 = r_4 + r_6$ (2.15)

Then

$$\left.\begin{aligned} r_6 &= r_6 \\ r_6 &= S_1 + r_5 \\ r_6 &= S_2 + r_4 \end{aligned}\right\} \qquad (2.16)$$

Now if not more than one error has occurred during transmission then it is possible to estimate both $S_1$ and $S_2$ at same time, i.e. first step. Once $S_1$ and $S_2$ have

... 23

been decoded correctly, we can use these results to estimate the single symbol $r_6$, i.e. second-step. Since the code is cyclic what can be done for $r_6$ can be done for all other $r_i$'s. Then, this code is said to be a two-step single error-correcting majority-logic decodable code.

In general, it is possible to define an L-step MLD code. The number of sets of equations orthogonal on certain received digits at each step till you reach a set orthogonal on a single received digit is dependent on the structure of the particular code [6] . If the code is completely orthogonalizable and if at each step each set of equations has D equations orthogonal on certain received digits then the code is said to be an L-step $t = \left[\frac{D-1}{2}\right]$ error-correcting MLD code.

L-Step Majority-Logic Decodable Codes.

Several classes of cyclic codes have been recently found to be L-step MLD. The construction and rules for orthogonalization of these codes are based on the properties of finite geometries [5] . To discuss finite geometries is beyond the scope of this thesis. In the following, only the parameters of these codes are given.

Let h be a non-negative integer less than $2^{ms}$,

... 24

where m and s are two positive integers. The integer h can be expressed in radix-$2^s$ form as follows:

$$h = a_0 + a_1 \cdot 2^{2s} + \ldots + a_{m-1} \cdot 2^{(m-1)s} \qquad (2.17)$$

where $0 \leq a_i < 2^s$ for $i = 0,1,2,\ldots,m-1$. Define the $2^s$-weight of h as the sum (ordinary sum) of the coefficients in its radix-$2^s$ expansion,

$$W_{2^s}(h) = \sum_{i=0}^{m-1} a_i \qquad (2.18)$$

consider the difference

$$h - W_{2^s}(h) = a_1(2^s-1) + \ldots + a_{m-1}(2^s-1)$$

It is obvious that h is divisible by $2^s-1$ if and only if $W_{2^s}(h)$ is divisible by $2^s-1$. Let $h^{(u)}$ be the remainder resulting from dividing $2^u-h$ by $2^{ms}-1$, i.e.

$$2^u \cdot h = q.(2^{ms}-1) + h^{(u)} \qquad (2.19)$$

(i) Finite Projective Geometry Codes (PG Codes [6]).

These codes are based on the properties of finite projective geometry. For any two positive integers m and s, there exists an L-step orthogonalizable code ($L < m$) with the following parameters:

$$\left.\begin{aligned} n &= (2^{ms}-1) / (2^s-1) \\ n-k &= \left[\begin{array}{l}\text{The number of non-negative integers h} \\ \text{less than } 2^{ms}-1 \text{ which are divisible by} \\ 2^s-1 \text{ and such that } \max_{0 \leq u < s} W_{2^s}(h^{(u)}) \\ = j(2^s-1) \text{ with } 0 \leq j < m-L\end{array}\right] \\ t &= \left[(2^{(m-L)}-1) / 2(2^s-1)\right] \end{aligned}\right\} (2.20)$$

... 25

For L = 1, we obtain a class of one-step MLD codes which contains the Difference-Set cyclic codes (m = 3) as a subclass. For L = 1 and s = 1, it yields the class of maximum length sequence codes.

(ii) Euclidean Geometry Codes (EG codes [6] ).

These codes are based on the properties of Euclidean geometry. An L-step MLD code of this class has the following parameters:

$$\left.\begin{aligned} n &= 2^{ms}-1 \\ n-k &= \left[\begin{array}{l}\text{The number of non-negative integers} \\ \text{less than } 2^{ms}-1 \text{ such that} \\ 0 \leq \max_{0 \leqslant u < s} W_{2s}(h^{(u)}) \leqslant (m-L)(2^s-1).\end{array}\right] \\ t &= \left[\frac{2^{(m-L+1)} \cdot s}{(2^s-1)} - 2\right] \end{aligned}\right\} \quad (2.21)$$

where m, s, and L are positive integers with L < m. For s = 1, we obtain a subclass of Euclidean geometry codes known as the Reed-Muller codes. The first MLD algorithm was devised in 1954 by Reed [17] for this class of codes discovered by Muller [18] .

Geometry codes were first studied by Rudolph [13] . Rudolph's work was later extended and generalized by many other coding investigators [19-27] .

In summary, we have given a definition of one-step and L-step majority-logic decoding. Some classes of codes which are MLD were also presented.

... 26

# CHAPTER 3

## NEW RESULTS.

### A - Preliminary Discussion.

Now, we give the material necessary in the discussion of the results to be presented later in this chapter.

Let $V_1$ be any binary code with minimum distance $d = 2t + 1$ and generated by $g_1(x)$. Then, every code word $V_1(x)$ can be expressed in the form

$$V_1(x) = g_1(x) \cdot I(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1} \qquad (3.1)$$

where

$$I(x) = \sum_{i=0}^{k-1} c_i x^i$$

is the information polynomial. As can be easily shown we can form starting with 3.1, the relations

$$\left.\begin{aligned} c_0 &= s_0, \\ c_0 &= s_1, \\ &\cdots \\ c_0 &= s_{d-1}, \end{aligned}\right\} \qquad (3.2)$$

where each $s_i$ is a sum of certain $r_j$'s. If each $r_j$ that occurs in these relations, occurs exactly once, then $V_1$ is one-step majority-logic decodable since t

errors can affect at most $\frac{d-1}{2}$ estimates of $C_0$ and, therefore at least $\frac{d+1}{2}$ estimates are bound to be correct. Also since $V_1$ is generated by a single generator polynomial, what is true about $C_0$ should be valid for other $C_i$'s. After estimating $C_0$, we can compute

$$x^{-1} \cdot (V_1(x) + C_0 \cdot g_1(x)) = r'_0 + r'_1 x + \ldots + r'_{n-2} x^{n-2} \qquad (3.3)$$

and using $r'_i$ instead of $r_i$ in 3.2, $C_1$ can be estimated. Clearly the procedure can be repeated to get $C_2, \ldots, C_{k-1}$ successively.

On the other hand suppose, starting with 3.1 we are able to construct the relations

$$\left.\begin{aligned} u &= s'_0 \\ u &= s'_1 \\ &\cdot \quad \cdot \\ &\cdot \quad \cdot \\ &\cdot \quad \cdot \\ u &= s'_{d-1} \end{aligned}\right\} \qquad (3.4)$$

and

$$\left.\begin{aligned} v &= s''_0 \\ v &= s''_1 \\ &\cdot \quad \cdot \\ &\cdot \quad \cdot \\ &\cdot \quad \cdot \\ v &= s''_{d-1} \end{aligned}\right\} \qquad (3.5)$$

... 28

where u is the sum of certain $C_i$'s, $v = u + C_{i_0}$ with $C_{i_0}$ not included in u, and $s'_j$ and $s''_j$ are like $s_j$ of 3.2. Then, we can estimate u by majority vote through 3.4. Using this estimate in 3.5 we can estimate $C_{i_0}$ by majority vote. Thus V would be two-step majority-logic decodable. With reference to 3.5, it is not necessary that u be present in all of the relations. Examples 8 and 9 illustrate this point.

If we require L sets of relations like 3.4 and 3.5 to estimate $C_{i_0}$, then V, would be L-step majority-logic decodable.

B - Some Useful Results on One-Step MLD Binary Codes.

Result 1:

Let V be an (n,k,t) binary one-step MLD code generated by g(x), then the (n,k,t) $V^*$ code generated by the reciprocal polynomial of g(x), i.e. $g^*(x) = x^{n-k} \cdot g(1/x)$, is also a one-step MLD code with the same error correcting capability as V.

Proof:

Let V(x) belong to V. Then $V(x) = g(x) \cdot I(x)$ (3.6)

or $r_0 + r_1x + \ldots + r_{n-1}x^{n-1} = (g_0 + g_1x + \ldots + g_{n-k}x^{n-k}) \cdot$

$(C_0 + C_1x + \ldots + C_{k-1}x^{k-1})$ (3.7)

... 29

Then also

$$\left.\begin{aligned} r_0 &= g_0 \cdot C_0 \\ r_1 &= g_1 \cdot C_0 + g_0 \cdot C_1 \\ r_2 &= g_2 \cdot C_0 + g_1 \cdot C_1 + g_0 \cdot C_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ r_{n-1} &= g_{n-k} \cdot C_{k-1} \end{aligned}\right\} \quad (3.8)$$

Now from 3.6

$$V^*(x) = g^*(x) \cdot I^*(c) \text{ belongs to } V^* \quad (3.9)$$

or

$$\left.\begin{aligned} r_0^* + r_1^* x + \ldots + r_{n-1}^* x^{n-1} &= (g_{n-k} + g_{n-k-1} x + \ldots + g_0 x^{n-k}) \cdot \\ &\quad (C_{k-1} + C_{k-2} x + \ldots + C_0 x^{k-1}) \\ &= (g_0^* + g_1^* x + \ldots + g_{n-k}^* x^{n-k}) \cdot \\ &\quad (C_0^* + C_1^* x + \ldots + C_{k-1}^* x^{k-1}) \end{aligned}\right\} \quad (3.10)$$

Then

$$\left.\begin{aligned} r_0^* &= g_{n-k} \cdot C_{k-1} \\ r_1^* &= g_{n-k} \cdot C_{k-2} + g_{n-k-1} \cdot C_{k-1} \\ &\cdot \\ &\cdot \\ &\cdot \\ r_{n-1}^* &= g_0 \cdot C_0 \end{aligned}\right\} \quad (3.11)$$

... 30

From 3.8 and 3.9 we get that

$$r_i^* = r_{n-1-i} \tag{3.12}$$

Now if one can form with respect to V sets of d equations orthogonal on each $C_i$, then with respect to $V^*$ one can form sets of d equations orthogonal on each $C^*_{k-1-i}$. Therefore $V^*$ is an $(n,k,t=\left[\frac{d-1}{2}\right])$ one-step MLD code Q.E.D.

Example 1:

With reference to the (15,7,2) cyclic code V generated by $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ one set of orthogonal estimates on $C_0$ is

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_4 + r_{12} + r_{13}, \\ C_0 &= r_2 + r_6 + r_{14}, \\ C_0 &= r_1 + r_3 + r_7, \\ C_0 &= r_8 + r_9 + r_{11}. \end{aligned}\right\} \tag{3.13}$$

Now for the (15,7,2) cyclic code $V^*$ generated by $g^*(x) = 1 + x + x^2 + x^4 + x^8$ one set of orthogonal estimates on $C^*_6$ is

$$\left.\begin{aligned} C^*_6 &= r^*_{14}, \\ C^*_6 &= r^*_{10} + r^*_2 + r^*_1, \\ C^*_6 &= r^*_{12} + r^*_8 + r^*_0, \\ C^*_6 &= r^*_{13} + r^*_{11} + r^*_7, \\ C^*_6 &= r^*_6 + r^*_5 + r^*_3. \end{aligned}\right\} \tag{3.14}$$

... 31

Result 2:

If V is an (n,k,t) binary L-step MLD code generated by g(x), then the (n,k) code $V^*$ generated by $g^*(x)$, the reciprocal polynomial of g(x), is also an L-step MLD code with the same error-correcting capability as V.

The proof follows directly from Result 1.

Result 3:

Let V be an (n,k,t) binary one step MLD code generated by g(x). Then the $(n^1 = n.i, k^1 = k.i)$ code $V^1$ generated by $g^1(x) = g(x^i)$ has the same error-correcting capability as V and is one-step MLD.

Proof:

Any V(x) belonging to V can be represented as

$$V(x) = g(x) \cdot I(x)$$
$$= (g_0+g_1x+\ldots+g_{n-k}x^{n-k}) \cdot (C_0+C_1x+\ldots+C_{k-1}x^{k-1}) \quad (3.15)$$
$$= r_0+r_1x+r_2x^2+\ldots+r_{n-1}x^{n-1}$$

From 3.15 we get

$$V(x^i) = g(x^i) \cdot (C_0 + C_1x^i + C_2x^{2.i} + \ldots + C_{k-1}x^{i.(k-1)}) \quad (3.16)$$

Note that V(x) and $V(x^i)$ have the same weight and that $V(x^i)$ belongs to $V^1$. Now suppose we choose for V(x) a word with minimum weight d. Then it follows that, if

... 32

$d^1$ is the minimum weight of $V^1$, we have

$$d^1 \not> d \qquad (3.17)$$

Next we show that $V^1$ can correct $\left[\frac{d-1}{2}\right]$ errors by one-step majority-logic decoding.

For every V(x) belonging to V given by 3.15, there is a $V^1(x)$ belonging to $V^1$ given by

$$V^1(x) = g(x^i) \cdot \left[C_0 + C_i x^i + C_{2.i} x^{2.i} + \ldots + C_{(k-1).i} x^{i.(k-1)}\right] = r_0 + r_i x^i + r_{2.i} x^{2.i} + \ldots + r_{i.(n-1)} x^{i.(n-1)} \qquad (3.18)$$

where $C_{p.i} = C_p$ and $r_{p.i} = r_p$. With reference to 3.15 we have a relation of the form

$$r_p = a_0.C_0 + a_1.C_1 + \ldots + a_{k-1}.C_{k-1} \qquad (3.19)$$

where $a_i$ and $C_j$ belong to GF(2). Multiplying the subscripts of r and C by i in 3.19 we get

$$r_{p.i} = a_0.C_0 + a_1.C_i + \ldots + a_{k-1}.C_{i.(k-1)} \qquad (3.20)$$

We observe that 3.20 is a relation relevant to 3.18. From this, it follows that if we can form d orthogonal estimates on $C_0$ with respect to 3.19, then we can also form d orthogonal estimates on $C_0$ with respect to 3.20. Further, since both V and $V^1$ are "cyclic", it can be concluded that if V is one-step MLD, then so is $V^1$.

... 33

However, it is not clear from the preceding discussion that in $V^1$, $n^1$ can be n.i rather than i.(n-1)+1. To clarify the point, we note with respect to 3.20, that, in forming the d orthogonal estimates on $C_0$, we do not use any r whose subscript is not divisible by i. This means that we can have $r_{i.(n-1)+1}, r_{i.(n-1)+2}, \ldots, r_{i.(n-1)+i-1}$. Thus $n^1$ is indeed n.i. Q.E.D.

Result 4:

Any code $V^1$ can correct by one step ML., the simultaneous occurrence of B bursts, each of length b.i or less, and $\left[\frac{d-1}{2}\right]$ -B.b random errors.

Proof:

The proof follows directly from the fact that a burst of length b.i can affect at most b equations of the type $C_{p.i} = \sum_j r_{p_j.i}$ and that B such bursts affect at most B.b relations. Q.E.D.

Result 5:

Any code $V^1$ can correct by 1SML the simultaneous occurrence of B bursts, each of length b.i + 1 or less, and $\left[\frac{d-1}{2}\right]$ -B.(b + 1) random errors. The proof follows directly from Result 4.

Result 6:

With reference to V, the (m.n,k) code $V_m$,

... 34

generated by $g_m(x) = g(x).\left[1 + x^n + x^{2.n} + x^{3.n} + \ldots + x^{(m-1).n}\right]$, has an error-correcting capability of $\left[\frac{m.d-1}{2}\right]$ and is one-step MLD.

The proof follows directly from Result 4.

Result 7:

The code $V'_m(m.n.i,\ k.i)$ generated by $g'_m(x) = g_m(x^i)$ can correct, by 1-SML, the simultaneous occurrence of B bursts, each of length b.i or less, and $\left[\frac{m.d-1}{2}\right]$ -B.b random errors.

The proof follows directly from Result 4.

Result 8:

The code $V'_m$ can correct, by 1-SML, the simultaneous occurrence of B bursts, each of length b.i + 1 or less, and $\left[\frac{m.d-1}{2}\right]$ -B.(b + 1) random errors.

The proof follows directly from Result 4.

Example 2:

The 1-SMLD (15,7,2) cyclic code V generated by $g(x) = 1 + x^4 + x^6 + x^7 + x^8$:

For this cyclic code one can form the five equations orthogonal on $C_0$:

$$\left.\begin{aligned} C_0 &= r_0 \\ C_0 &= r_4 + r_{12} + r_{13} \\ C_0 &= r_1 + r_3 + r_7 \\ C_0 &= r_2 + r_6 + r_{14} \\ C_0 &= r_8 + r_9 + r_{11} \end{aligned}\right\} \qquad (3.13)$$

... 35

This code can correct two random errors. For the code V' (75,35,2) generated by $g'(x) = g(x^5)$ one can form the following five equations orthogonal on $C_0$:

$$\left.\begin{aligned} C_0 &= r_0 \\ C_0 &= r_{20} + r_{60} + r_{65} \\ C_0 &= r_5 + r_{15} + r_{35} \\ C_0 &= r_{10} + r_{30} + r_{70} \\ C_0 &= r_{40} + r_{45} + r_{55} \end{aligned}\right\} \qquad (3.21)$$

This code V' has the same rate as V, i.e. $R = \frac{7}{15}$, and can also correct either two random errors, or one burst of length 5 and one random error, or two bursts of length 5, or one burst of length 10. The sacrifice is an increase in length from $n = 15$ to $n = 75$.

C - A Class of Single Error Correcting Codes.

Let V be a binary (n,k) code generated by $g(x) = 1 + x^2 + x^{2a+1}$ where $n = (2a+1)(a+1)$, $k = (2a+1)a$, $a \geqslant 1$. Then V is a one-step majority-logic decodable single error-correcting code.

Proof:

Since g(x) has weight 3, the minimum distance of V is $\leqslant 3$. Therefore, it is sufficient to show that, with reference to 3.2, we can form 3

... 36

estimates on $C_0$ such that every $r_j$ occurring will occur exactly once.

Consider the tree of numbers, shown in Fig. 3.1, starting with the number $2(2a+1) = 4a+2$. From this number we get the numbers $(4a+2) + (2a+1) = 6a+1$ and $(4a+2) + (2a+1) = 6a+3$. From $6a+1$ we get $(6a+1) + (2a-1) = 8a$ and $(6a+1) + (2a+1) = 8a+2$. From $6a+3$ we get $(6a+3) + (2a-1) = 8a+2$ and $(6a+3) + (2a+1) = 8a+4$. Since $8a+2$ occurs twice we remove it and do not consider any numbers arising from this. In general we can say that from every number N occurring we get two numbers, $N + (2a-1)$ and $N + (2a+1)$. If a number occurs twice we remove it and do not use it in getting further numbers. For a reason to be made clear later, we wish to find at what stage we get two numbers which differ by one. Examination of Fig. 3.1 shows that to get these two numbers we have to consider only the top-most and bottom-most paths. The numbers on the top-most path are all of the form $(4a+2) + p(2a-1)$ and those on the bottom-most path are of the form $(4a+2) + q(2a+1)$. To get the two smallest numbers which differ by one we have to consider

$$(4a+2) + p(2a-1) = (4a+2) + (p-1)(2a+1) + 1$$

... 37

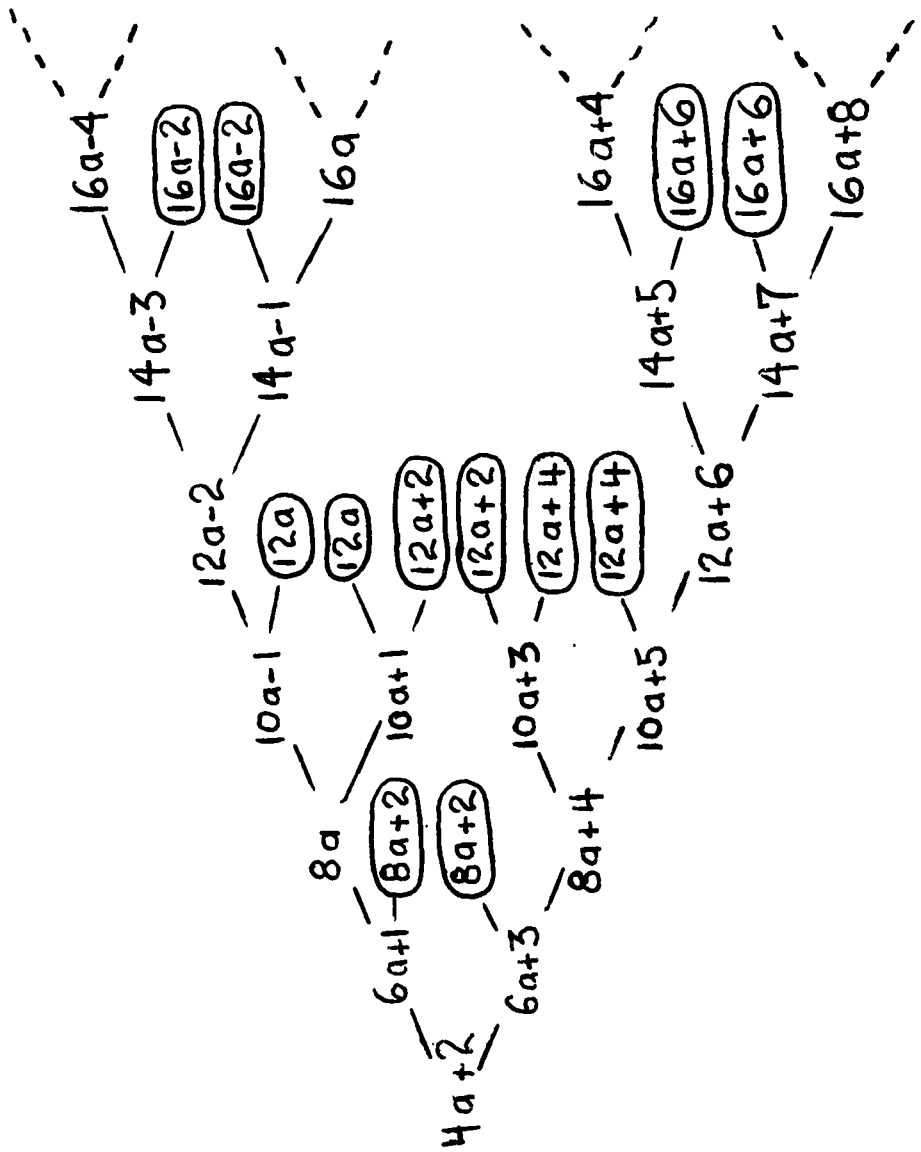
16a-4
16a-2
16a-2
16a
16a+4
16a+6
16a+6
16a+8
14a-3
14a-1
14a+5
14a+7
12a-2
12a
12a
12a+2
12a+2
12a+4
12a+4
12a+6
10a-1
10a+1
10a+3
10a+5
8a
8a+2
8a+2
8a+4
6a+1
6a+3
4a+2

FIGURE 3.1

$$\text{or} \qquad p\left[2a-1-(2a+1)\right] = -(2a+1)+1 \qquad (3.22)$$

$$\text{or} \qquad p = a.$$

Therefore, the two required numbers are

$$(a+1)\,.\,(2a+1) \quad \text{and} \quad (a+1)\,.\,(2a+1)+1.$$

Let $S_1$ be the set of all numbers, from Fig. 3.1 which are less than $(a+1)\,.\,(2a+1)-1 = n-1$ where n is the code length. Let $S_2$ be the set consisting of the number $n-1$ and all those numbers obtained by taking each number of $S_1$ and subtracting one from it. Clearly $S_1$ and $S_2$ are disjoint if $n = (a+1)\cdot(2a+1)$. Now let us consider

$$\begin{aligned} s_0 &= r_0, \\ s_1 &= r_1 + r_3 + \ldots + r_{2a+1} + \textstyle\sum_i r_i, \ i \text{ belongs to } S_1, \\ s_2 &= r_2 + r_4 + \ldots + r_{2a} + \textstyle\sum_j r_j, \ j \text{ belongs to } S_2. \end{aligned} \qquad (3.23)$$

We have $s_0 = s_1 = s_2 = C_0$, since

$$r_i = \begin{cases} C_i & \text{for } 0 \leqslant i \leqslant 1, \\ C_{i-2} + C_i & \text{for } 2 \leqslant i \leqslant 2a, \\ C_{i-2a-1} + C_{i-2} + C_i & \text{for } 2a+1 \leqslant i \leqslant n-1, \end{cases}$$

where $C_i = 0$ for $i \geqslant a(2a+1)$.

... 39

Also since $S_1$ and $S_2$ are disjoint for $n = (a+1)(2a+1)$, the set 3.23 is orthogonal on $C_0$. Thus the code V is single error-correcting and majority-logic decodable in one step, with $n = (a+1)(2a+1)$, $k = a.(2a+1)$ and rate $R = a/(a+1)$.

It can be shown that more generally the $(n' = (Aa+B).(a+1), k' = a(Aa+B)$ codes generated by $1 + x^A + x^{Aa+B}$, where $A \geq 2$, $1 \leq B \leq A-1$ and $a \geq 1$, are again single error-correcting and one-step majority-logic decodable. But they are not good in the sense that for the same rate, they are longer than V.

Clearly V has rate $= \frac{k}{n} = \frac{(2a+1)a}{(2a+1)(a+1)} = \frac{a}{a+1}$.

Table I gives the values of n, k and the rate for a up to 17.

TABLE I

| a | n | k | Rate |
|---|---|---|---|
| 1 | 6 | 3 | 1/2 |
| 2 | 15 | 10 | 2/3 |
| 3 | 28 | 21 | 3/4 |
| 4 | 45 | 36 | 4/5 |
| 5 | 66 | 55 | 5/6 |
| 6 | 91 | 78 | 6/7 |
| 7 | 120 | 105 | 7/8 |
| 8 | 153 | 126 | 8/9 |
| 9 | 190 | 171 | 9/10 |
| 10 | 231 | 210 | 10/11 |
| 11 | 276 | 253 | 11/12 |
| 12 | 325 | 300 | 12/13 |
| 13 | 378 | 351 | 13/14 |
| 14 | 435 | 406 | 14/15 |
| 15 | 496 | 465 | 15/16 |
| 16 | 561 | 528 | 16/17 |
| 17 | 630 | 595 | 17/18 |

... 40

These values are identical with those of TW codes [3] for single error correction. This can be easily established as follows: In the notation of Townsend and Weldon [3] , each disjoint difference set has order d-1 = 2 and the number of disjoint difference sets is $k_0$. Therefore the total number of differences is $(d-2).(d-1).k_0 = 2.k_0$ and this should be $\leq m-1$. Thus $n = m.n_0 \geq (2.k_0 + 1).n_0$. Noting that $n_0 = k_0 + 1$, we get $n \geq (2.k_0+1).(k_0+1)$. However, for the case of $d = 3$, the disjoint sets are $(0,1),(0,2),\ldots(0,k_0)$ and the differences from these sets, when taken modulo $m = 2.k_0 + 1$, are clearly $1,2,\ldots m-1$. Thus in the present case $n = (2.k_0 + 1).(k_0 + 1)$. But $k_0$ is equal to our a. Therefore $n = (2a + 1).(a + 1)$ which is also the length of V. From this it also follows that V and TW codes have the same k since their rates are equal. An implication of the preceding discussion is that the class of V is as large as the class of TW codes for single error correction.

If the exponent of g(x) is n', then g(x) generates a single error-correcting code of length n'. Thus V can be treated as a shortened cyclic code.

Also, if the prescribed length is not according to $n = (2.a + 1).(a + 1)$, then the V

... 41

with next longer length can be taken and shortened to the prescribed length.

Example 3:

For $a = 5$, we have $g(x) = 1 + x^2 + x^{11}$ , and V is (66,55) giving a rate of 5/6. The tree of numbers, relevant to Fig. 3.1, is shown in Fig. 3.2 .

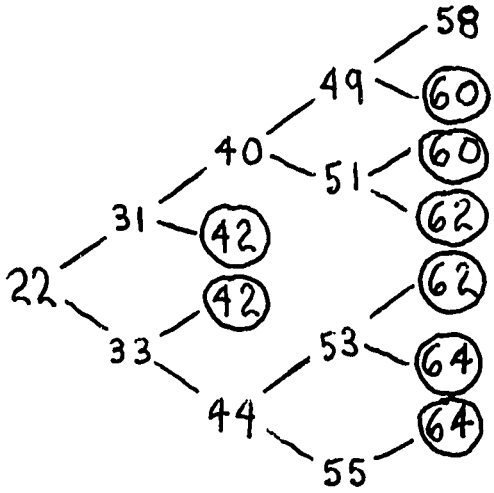


Fig. 3.2

With reference to 3.23 and Fig. 3.2 , we have

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_1 + r_3 + r_5 + r_7 + r_9 + r_{11} + r_{22} + r_{31} + \\ &\quad r_{33} + r_{40} + r_{44} + r_{49} + r_{51} + r_{53} + r_{55} + \\ &\quad r_{58}, \\ C_0 &= r_2 + r_4 + r_6 + r_8 + r_{10} + r_{21} + r_{30} + r_{32} + \\ &\quad r_{39} + r_{43} + r_{48} + r_{50} + r_{52} + r_{54} + r_{57} + \\ &\quad r_{65}. \end{aligned}\right\} \quad (3.24)$$

... 42

Example 4:

For $a = 2$, we have $g(x) = 1 + x^2 + x^5$, and V is (15,10). With reference to 3.23 and Fig. 3.1, we have

$$\left.\begin{aligned} c_0 &= r_0, \\ c_0 &= r_1 + r_3 + r_5 + r_{10} + r_{13}, \\ c_0 &= r_2 + r_4 + r_9 + r_{12} + r_{14}. \end{aligned}\right\} \qquad (3.25)$$

The code has rate 2/3.

Example 5:

For $a = 3$, we have $g(x) = 1 + x^2 + x^7$, and V is (28, 21). With reference to 3.23 and Fig. 3.1 we have

$$\left.\begin{aligned} c_0 &= r_0, \\ c_0 &= r_1 + r_3 + r_5 + r_7 + r_{14} + r_{19} + r_{21} + r_{24}, \\ c_0 &= r_2 + r_4 + r_6 + r_{13} + r_{18} + r_{20} + r_{23} + r_{27}. \end{aligned}\right\} \qquad (3.26)$$

The code has rate 3/4.

Example 6:

For $a = 1$, we have $g(x) = 1 + x^2 + x^3$ and V is (6,3). With reference to 3.23 and Fig. 3.1 we have

$$\left.\begin{aligned} c_0 &= r_0, \\ c_0 &= r_1 + r_3, \\ c_0 &= r_2 + r_5. \end{aligned}\right\} \qquad (3.27)$$

This code has rate 1/2.

... 43

Three Lengthened Codes.

It is possible to lengthen V to make it majority logic decodable in more than one step. We do not have any general rules regarding this aspect of the problem. However, the following three specific two-step cases may prove to be of some interest.

Example 7:

The code of Example 4 can be lengthened to (19,14) giving a rate of $\frac{14}{19} \simeq 0.737$ as against the rate of $2/3 \simeq 0.67$ of the (15,10) parent code. The relations, parallel to 3.25 would now be

$$\left.\begin{aligned} C_0 + C_{10} + C_{13} &= r_0 + r_{15}, \\ C_0 + C_{10} + C_{13} &= r_1 + r_3 + r_5 + r_{10} + r_{13} + r_{16}, \\ C_0 + C_{10} + C_{13} &= r_2 + r_4 + r_9 + r_{12} + r_{14} + r_{18}, \end{aligned}\right\} \quad (3.28)$$

and

$$\left.\begin{aligned} C_0 + C_{10} + C_{13} + C_5 &= r_0 + r_{10} + r_{13} + r_{16}, \\ C_0 + C_{10} + C_{13} + C_5 &= r_2 + r_7 + r_{12} + r_{17} + r_{18}, \\ C_0 + C_{10} + C_{13} + C_5 &= r_1 + r_3 + r_5 + r_{15}. \end{aligned}\right\} \quad (3.29)$$

Note that by using the estimate of $C_0 + C_{10} + C_{13}$, obtained through 3.28 in 3.29, $C_5$ is estimated.

Since the (31,26) cyclic code can also be generated by $1 + x^2 + x^5$, the (19,14) lengthened code is a shortened cyclic code.

... 44

Example 8:

The code of Example 5 can be lengthened to (34,27) giving a rate of $\frac{27}{34} \simeq 0.794$ as against the rate of $\frac{3}{4} = 0.75$ of the (28,21) parent code. The relations, parallel to 3.26, would be

$$\left.\begin{aligned} C_0 + C_{21} &= r_0 + r_{28} + r_{33}, \\ C_0 + C_{21} &= r_1 + r_3 + r_5 + r_7 + r_{14} + r_{19} + r_{21} + r_{24} + r_{29} + r_{31}, \\ C_0 + C_{21} &= r_2 + r_4 + r_6 + r_{13} + r_{18} + r_{20} + r_{23} + r_{27} + r_{30} + r_{32}, \end{aligned}\right\} \quad (3.30)$$

and

$$\left.\begin{aligned} C_{20} &= r_{27} + r_{32}, \\ C_0 + C_{20} + C_{21} &= r_2 + r_4 + r_6 + r_{13} + r_{18} + r_{20} + r_{23} + r_{30}, \\ C_{20} &= r_1 + r_3 + r_5 + r_{12} + r_{17} + r_{19} + r_{22} + r_{26} + r_{29} + r_{31} + r_{33}. \end{aligned}\right\} \quad (3.31)$$

Note that by using the estimate of $C_0 + C_{21}$, obtained through 3.30, in 3.31, $C_{20}$ can be estimated.

Example 9:

The code of Example 6 can be lengthened to (8,5) giving a rate of $\frac{5}{8} = 0.625$ as against the rate of $\frac{1}{2} = 0.500$ of the (6,3) parent code. The relations, parallel to 3.27, would be

$$\left.\begin{aligned} C_0 + C_3 &= r_0 + r_6 + r_7, \\ C_0 + C_3 &= r_2 + r_5, \\ C_0 + C_3 &= r_1 + r_3, \end{aligned}\right\} \quad (3.32)$$

and

... 45

$$\left.\begin{aligned} c_1 &= r_1, \\ c_0 + c_1 + c_3 &= r_3, \\ c_0 + c_1 + c_3 &= r_2 + r_4 + r_6. \end{aligned}\right\} \qquad (3.33)$$

These codes when used with the results of section B are found to compare favourably with the Fire Codes [4,5,6] for short burst error correction.

In this chapter, some useful results on one-step majority-logic decodable codes and a class of single error-correcting majority-logic decodable codes were presented. This class was shown to be as large as the class of TW [3] codes for single error correction. In fact, the values of n and k were found to be identical with those of TW codes. However, V is generated by a single polynomial as in the case of cyclic codes, and is, therefore, somewhat simpler to implement than TW codes. No general rules have been found to lengthen V to make it majority-logic decodable in more than one step. However, we have given three examples for the two-step case.

... 46

## CONCLUSION

We have presented some useful results on one-step MLD binary codes and a class of binary single error-correcting codes V which are generated by $g(x) = 1 + x^2 + x^{2a+1}$ and are one-step majority-logic decodable. It turns out to be that both V and TW codes [3] for single error correction have the same values for n and k. In this connection the following specific codes may prove to be of interest.

The (10,5) code generated by $1 + x^2 + x^3 + x^5$ has minimum distance $d = 4$ and the relations 3.2 in this case are

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_1 + r_3 + r_8, \\ C_0 &= r_2 + r_7 + r_9, \\ C_0 &= r_4 + r_5 + r_6. \end{aligned}\right\} \qquad (4.1)$$

This code is interesting in that the length of a self-orthogonal code [3] with $d = 4$ and rate $\frac{1}{2}$ must be at least 14. An exhaustive examination shows that there is no (8,4) one-step majority-logic decodable code, generated by a single polynomial and having $d = 4$.

The (16,9) code generated by $1 + x^2 + x^5 + x^7$ compares favourably with the (14,7) TW code for $d = 4$. Here the relations 3.2 are

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_1 + r_3 + r_5 + r_{12} + r_{14}, \\ C_0 &= r_2 + r_4 + r_{11} + r_{13} + r_{15}, \\ C_0 &= r_6 + r_7 + r_8 + r_9 + r_{10}. \end{aligned}\right\} \qquad (4.2)$$

The (34,21) code generated by $1 + x^2 + x^{11} + x^{13}$ has $d = 4$ and has for relations 3.2,

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_1 + r_3 + r_5 + r_7 + r_9 + r_{11} + r_{24} + r_{26} + r_{28} + r_{30} + r_{32}, \\ C_0 &= r_2 + r_4 + r_6 + r_8 + r_{10} + r_{23} + r_{25} + r_{27} + r_{29} + r_{31} + r_{33}, \\ C_0 &= r_{12} + r_{13} + r_{14} + \cdots + r_{22}, \end{aligned}\right\} \qquad (4.3)$$

whereas the corresponding TW code is (39,26).

Note that for even d not only is it possible to correct $t = \left[\frac{d-1}{2}\right]$ but it is also possible to detect other errors since for d even it is possible to have $\frac{d}{2}$ estimates "0" and $\frac{d}{2}$ estimates "1".

The (26,13) code generated by $1 + x^2 + x^5 + x^7 + x^{13}$ has $d = 5$ and has for relations 3.2

$$\left.\begin{aligned} C_0 &= r_0, \\ C_0 &= r_2 + r_{15} + r_{21} + r_{23}, \\ C_0 &= r_5 + r_{16} + r_{18} + r_{22}, \\ C_0 &= r_4 + r_7 + r_{12} + r_{17}, \\ C_0 &= r_3 + r_8 + r_{13} + r_{24}. \end{aligned}\right\} \qquad (4.5)$$

... 48

From these results, we offer as a conjecture that it might be possible in specific cases to obtain generator polynomials for the multiple error-correcting cases although we have not been able to obtain any results in this regard.

... 49

REFERENCES.

1. C.E. Shannon, and W. Weaver, "A Mathematical Theory of Communication", University of Illinois Press, Urbana, Illinois, 1949.

2. C.E. Shannon, "Certain Results in Coding Theory for Noisy Channels", Information and Control, 1, September, 1957, pp. 6-25.

3. R.L. Townsend, and E.J. Weldon, Jr., "Self-Orthogonal Quasicyclic Codes", IEEE Trans. IT-13, No. 2, April, 1967, pp. 183-195.

4. W.W. Peterson, "Error-Correcting Codes", The M.I.T. Press, Cambridge, Massachusetts, and John Wiley, New York, 1961.

5. E.R. Berlekamp, "Algebraic Coding Theory", McGraw-Hill, New York, 1968.

6. Shu Lin, "An Introduction to Error-Correcting Codes", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

7. James L. Massey, "Threshold Decoding", The M.I.T. Press, Cambridge, Massachusetts, 1963.

8. _____ , "Advances in Threshold Decoding" in Advances in Communications Systems, Vol. 3, A.V. Balakrishnan, Ed., Academic Press, New York, 1968, pp. 91-115.

... 50

9. E.J. Weldon Jr., "Some Results on Majority-Logic Decoding", Chapter 8, Error Correcting Codes, H. Mann, Ed., John Wiley, 1968.

10. R.B. Yale, "Error Correcting Codes and Linear Recurring Sequences", Lincoln Laboratory Report 34-37, Lincoln Labs., M.I.T., 1958.

11. N. Zierler, "On A Variation of the First Order Reed-Muller Codes", Lincoln Laboratory Report 34-80, Lincoln Labs., M.I.T., 1958.

12. J. Singer, "A Theorem in Finite Projective Geometry and Some Applications to Number Theory", AMS Trans., 43, 1938, pp. 377-385.

13. L.D. Rudolph, "Geometric Configuration and Majority-Logic Decodable Codes", MEE Thesis, University of Oklahoma, Norman, Oklahoma, 1964.

14. _____ , "A Class of Majority-Logic Decodable Codes", IEEE Trans., IT-13, No. 2, April, 1967, pp. 305-307.

15. E.J. Weldon Jr., "Difference-Set Cyclic Codes", Bell Systems Techn. J., 45, September, 1966, pp. 1045-1055.

16. F.L. Graham and J. McWilliams, "On the Number of Parity Checks in Difference-Set Cyclic Codes", Bell Systems Techn. J., 45, September, 1966, pp. 1056-1070.

... 51

17. I.S. Reed, "A Class of Multiple Error-Correcting Codes and the Decoding Scheme", IRE Trans., IT-4, September, 1954, pp. 38-49.

18. D.E. Muller, "Applications of Boolean Algebra to Switching Circuit Design and to Error Detection", IRE Trans., EC-3, September, 1954, pp. 6-12.

19. J.M. Goethals, and P. Delsarte, "On a Class of Majority-Logic Decodable Codes", IEEE Trans., IT-14, March, 1968.

20. T. Kasami, S. Lin, and W.W. Peterson,"New Generations of the Reed-Muller Codes-Part I: Primitive Codes", IEEE Trans., IT-14, March, 1968.

21. _____, "Polynomial Codes", IEEE Trans., IT-14, November, 1968, pp. 807-814.

22. S. Lin, "On a Class of Cyclic Codes", Chapter 7, Error-Correcting Codes, H. Mann, Ed., John Wiley, New York, 1968.

23. W.W. Peterson, and E.J. Weldon, Jr., "Error-Correcting Codes", 2nd Edition, The M.I.T. Press, Cambridge, Massachusetts, 1970.

24. K.J.C. Smith, "Majority Decodable Codes Derived from Finite Geometries", Institute of Statistics Mimeo Series No. 561, University of North Carolina, Chapel Hill, North Carolina, 1967.

... 52

25. E.J. Weldon Jr., "Euclidean Geometry Cyclic Codes", Proceedings of the Symposium of Combinatorial Mathematics at the University of North Carolina, Chapel Hill, North Carolina, April, 1967.

26. E.J. Weldon Jr., "New Generations of the Reed-Muller Codes-Part II: Non-primitive Codes", IEEE Trans., IT-14, March, 1968, pp. 199-205.

27. T. Kasami, and Shu Lin, "On Majority-Logic Decoding for Duals of Primitive Polynomial Codes", IEEE Trans., IT-17, No. 3, May, 1971, pp. 322-331.

## BIOGRAPHY

Name: Richard Provost

Date of Birth: September 20, 1946

Birth Place: Ottawa, Ontario, Canada

Education: 1) University of Ottawa
High School (1963)

2) B.A.Sc. from the
University of Ottawa (1969)