CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL 339 (r. 86/09)

ON FINDING GENERATOR POLYNOMIALS AND PARITY-CHECK SUMS OF BINARY PROJECTIVE GEOMETRY CODES

by

Huang, Jen-Fa

A thesis
submitted to the School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

University of Ottawa
Ottawa, Canada
April, 1985

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ABSTRACT

Finding generator polynomials and parity-check sums for encoding and decoding binary projective geometry codes is considered in this thesis. The generator polynomials are found through the determination of coset representatives. The parity-check sums are found through the construction of subspaces in projective geometries.

Let V be an (n,k,d) binary projective geometry code with length $n=(q^{m+1}-1)/(q-1)$, $q=2^s$, and minimum distance $d \geq [(q^{m-r+1}-1)/(q-1)]+1$. This code is r-step majority-logic decodable. With reference to $GF(q^{m+1})=\{0,1,\alpha,\alpha^2,\ldots,\alpha^{n(q-1)-1}\}$, the generator polynomial g(X) of V has α^{ν} as a root if and only if ν has the form $i(q-1)$ and $\max_{0\leq \ell<s} W_q(2^{\ell}\nu) \leq (m-r)(q-1)$, where $W_q(2^{\ell}\nu)$ indicates the radix-q weight of the number $2^{\ell}\nu$.

Regarding the above condition, we note that if ν is an acceptable number, then so is every number belonging to the cyclotomic coset containing ν. This

suggests that if we can find a coset representative from each acceptable coset, then we can obtain g(X) easily. By using this line of attack, we determine the coset representatives for the cases of m-r=1, with any s, and m-r=2, with $s \geq 2$.

On the decoding side of V, we note that for any (r-1)-flat and any point not belonging to it, there is a unique r-flat in PG(m,q) which contains both the (r-1)-flat and the point. By recursively applying this for r=1,2,...,m-1, we determine the r-flats in PG(m,q) that intersect on a given (r-1)-flat in PG(m,q). Furthermore, at each level of orthogonalization, we examine the generating orthogonal flat f(X) such that every flat in the orthogonal check set can be expressed as a multiple of f(X) and hence the sequential-code-reduction decoding can be implemented.

ACKNOWLEDGEMENTS

The author is deeply indebted to the invaluable advice and continuous encouragement received from his supervisor Professor S.G.S. Shiva, without whom the work reported in this thesis would not have been possible.

The author is grateful to Dr. G. Seguin of the Royal Military College in Kingston, for his stimulating discussions. To Mr. John Jones of University of Ottawa, a special "thanks" for his careful proofreading of the final manuscript.

Thanks are also due to various members of the staff of the Electrical Engineering Department for their assistance on typesetting and/or computer operations.

Finally, the author wishes to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada under grant A-3371 and from the University of Ottawa in the form of a graduate assistantship.

TABLE OF CONTENTS

NOMENCLATURE

$GF(q)$:	Galois field with q elements;
$GF(q^{m+1})$:	Extended Galois field of dimension m+1 and over the prime field GF(q);
$PG(m,q)$:	m-dimensional projective geometry over GF(q);
$\deg P(X)$:	The degree of polynomial P(X);
$P^*(X)$:	The reciprocal polynomial of P(X);
$P(X) \mid Q(X)$:	P(X) divides Q(X);
α :	Primitive element of GF(q);
β :	Primitive element of $GF(q^{m+1})$;
C_i :	Cyclotomic coset having coset element i;
$M_i(X)$:	Minimal polynomial of field element α^i;
$g(X)$:	Generator polynomial;
$h(X)$:	Parity-check polynomial;
$E(X)$:	Idempotent polynomial;
$f(X)$:	Incidence polynomial of r-flat $F^{(r)}$; also, the generating orthogonal flat;
$C_j(X)$:	The jth tap-coefficient polynomial used in sequential-code-reduction decoding;
Z_t :	Set of integers {0, 1, 2, ..., t-1};
S_ν :	Set of nonzero numbers ν such that α^ν is a root of g(X);
(α^i) :	A point (0-flat) in PG(m,q);
$F^{(r)}$:	An r-flat in PG(m,q);

$W_q(\nu)$:	The weight of the radix-q representation of the number ν;
$\max_{0\le\ell<s} W_q(2^\ell\nu)$:	The maximum of the radix-q weights $W_q(\nu)$, $W_q(2\nu)$, ..., $W_q(2^{s-1}\nu)$;
GCD(x,y) :	The Greatest Common Divisor of x and y, usually abbreviated as (x,y);
$[x/y]$:	The greatest integers less than or equal to x/y;
$\begin{bmatrix} x \\ y \end{bmatrix}$	q-ary Gaussian binomial coefficient, defined as $\begin{bmatrix} x \\ y \end{bmatrix} = \prod_{j=0}^{y-1} \frac{q^x - q^j}{q^y - q^j}$;
$\binom{x}{y}$ or C(x,y) :	y-combinations of x objects, defined as $\binom{x}{y} = \frac{x!}{y!\,(x-y)!}$;
$x \in S_x$:	Element x belongs to the set S_x;
$x \equiv y \bmod z$:	y is congruent to x with operation taken modulo z;
$\langle i_0, i_1, \ldots, i_t \rangle$:	The abbreviated designation of the number $2^{i_0} + 2^{i_1} + \ldots + 2^{i_t}$;
$\{i_0, i_1, \ldots, i_t\}$:	A set consists of elements i_0, i_1, ..., i_t;
$F_1^{(r)} \cup F_2^{(r)} \cup \ldots \cup F_t^{(r)}$:	Union of r-flats $F_1^{(r)}$, $F_2^{(r)}$, ..., and $F_t^{(r)}$;
$F^{(r)} - F^{(r-1)}$:	Difference set of r-flat $F^{(r)}$ and (r-1)-flat $F^{(r-1)}$;
(n,k) code :	A linear code with codeword length n and number of information symbols k;
(r,s)th- PG code :	The largest cyclic code whose dual space contains all the incidence vectors of the r-flats in $PG(m,p^s)$.

Chapter I

INTRODUCTION

1.1 BASIC CONCEPT OF ERROR CONTROL CODING

Communication links transmit information from here to there. Computer memories transmit information from now to then. In either case, noise causes the received data to differ slightly from the original data. Error control coding enables a system to achieve a high degree of reliability despite the presence of noise. In addition to the data bits one wishes to transmit, one also transmits some additional redundant check bits. Even though the noise causes some errors in both the transmitted data bits and the transmitted check bits, there is usually still enough information available to the receiver to allow a sophisticated decoder to correct all of the errors unless the noise is extremely severe.

The reasons for the increasing applications of coding are numerous: the rapid growth in satellite communications; the revolution in digital integrated

circuits with large scale integration; the increasing emphasis on the reliable transmission of digital data and of digitally coded analog signals; the availability of relatively inexpensive computers for system, algorithm, and hardware simulation; the increasing digitalization of modems, switches, and interconnected facilities; and the increasing sophistication of the user community.

Developments within the coding field are equally important. The discoveries of new classes of codes and the inventions of powerful decoding algorithms have been critical to the significant application of coding theory.

The main thrust of coding theory has been directed towards finding codes that satisfy Shannon's noisy coding theorem [40]. The theorem states that the probability of transmission error can be made arbitrarily small by suitable coding procedures as long as the transmission rate of information is less than the channel capacity. The rate can be reduced to accommodate the channel capacity by

adding redundancy to the information prior to its transmission, and removing it at the receiver. This redundancy is incorporated in a structured fashion such that a given information sequence always results in the identical redundancy being added to it by the channel encoder.

The block diagram shown in Figure 1.1 illustrates the basic elements of a digital communications system employing coded waveforms.

The purpose of the channel encoder is to introduce some structured redundancy to the source output I(X), on the basis of a one-to-one assignment rule that produces a channel codeword V(X). The added redundancy is used by the channel decoder in its attempt to remove the errors introduced by the noisy channel.

The binary digits from the encoder are fed into a modulator which maps each bit into an elementary signal waveform. The channel over which the waveforms are transmitted is a spatial link between the source and the sink such as a cable or a satellite link. It can also

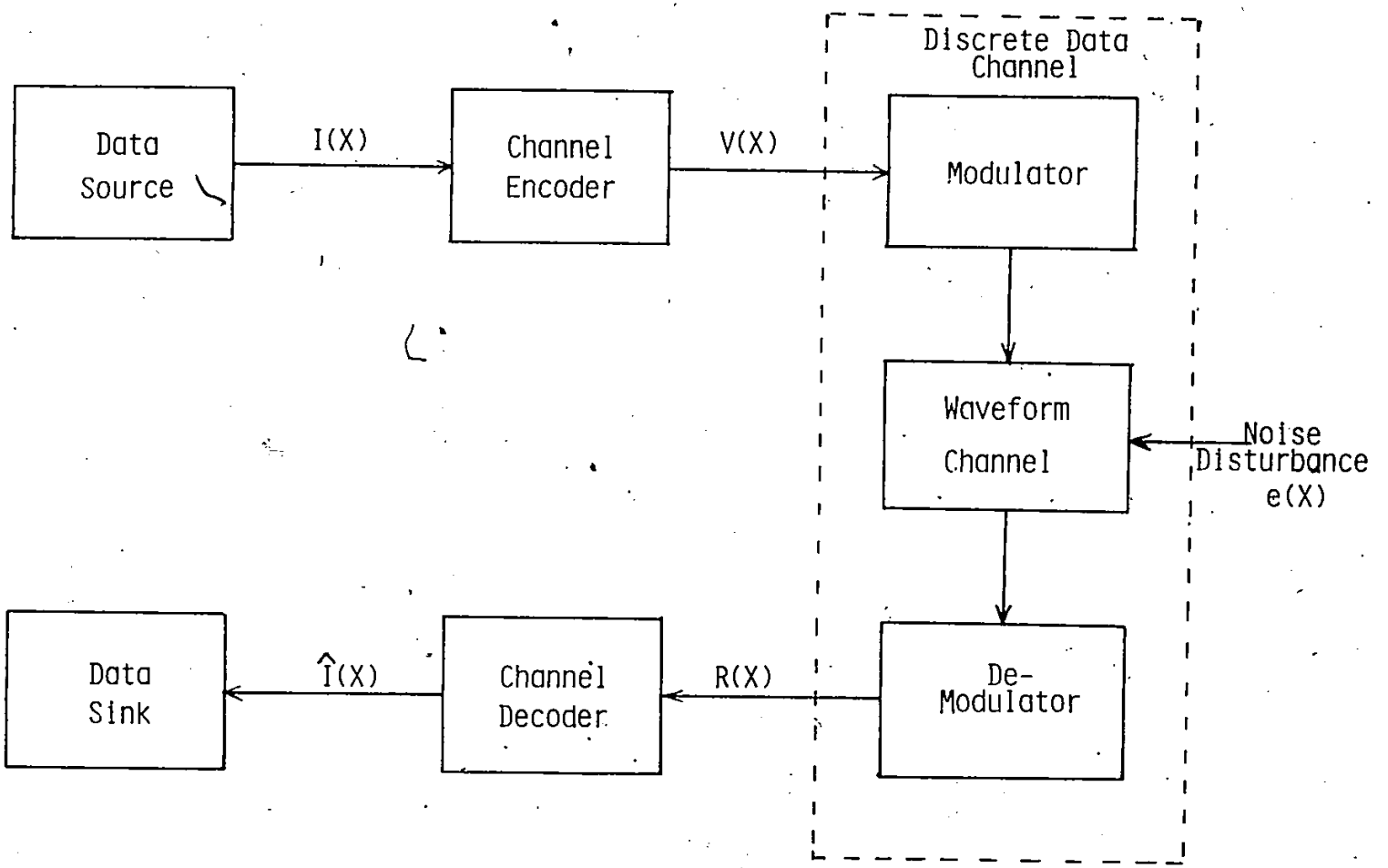


Figure 1.1 General model of digital communication process.

be a storage medium such as a magnetic tape, disc, or semiconductor memory. In all cases, the channel has the capability of corrupting the information being transmitted or stored.

On the receiver side, the demodulator can be viewed as the matched filter to the signal waveform corresponding to each transmitted bit. Its output is quantized to bit sequence and is fed into the channel decoder: The channel decoder can detect or correct the errors caused by the channel, as long as the number of errors per codeword does not exceed the error-detecting or error-correcting capability of the code. The function of the decoder is to categorize the many possible channel outputs into a generally much smaller number of channel decoder outputs. The decoder output is $\hat{I}(X)$, an estimate of the channel encoder input $I(X)$.

Error-control codes [2], [23] are used for the correction or detection of errors occurring during transmission over a noisy channel. For the Binary Symmetric Channel (BSC) model considered above, the effect of errors occurring during transmission can be described by the addition of an error-vector $e(X)$ to the

transmitted code vector V(X), resulting in the reception of the vector R(X)=V(X)+e(X), where addition is performed in GF(2). The extent by which code sequences differ from each other and from the received sequence, is measured by the Hamming distance which is defined as the number of dissimilar bit positions. Associated with this is the definition of Hamming weight, which is the number of "one" bits in a given sequence. On this basis the Hamming distance between the transmitted vector and the received vector equals the Hamming weight of the error vector.

With the rapidly increasing range and variety of channel coding techniques, there has been a correspondingly increasing need for systems-designers to assess the benefits of these techniques quantitatively, both for comparison of different codes and to gauge whether, in given circumstances, coding should be used at all. The concept of Net Coding Gain [4], [7] has been used to express the improvement attained by coding in terms of effective signal-to-noise ratio so that the effect of coding can be compared with other expedients, such as increasing the energy per bit. For a constant information rate, the modulation rate has to be

increased if coding is used. This generally leads to reduced energy per bit, and hence a higher error rate during transmission than for the uncoded case. The comparison to be made is thus between an unprotected system transmitting k bits/sec, and a coded system transmitting n bits/sec, where n > k. The coded system has a higher channel-error rate, but, if adequately designed, should more than compensate for this by the action of the decoder, so that a positive gain can be achieved.

There have been two fundamentally different approaches to the error-control coding problem, namely block coding and convolutional coding. In block coding, the channel encoder partitions the continuous sequence of information digits into k-symbol blocks. It then encodes each block independently, according to the particular code to be employed, and producing an n-symbol code word containing n-k parity check symbols. This n-symbol block is then transmitted over the channel, corrupted by noise, and decoded independently of all other code words. The code which maps a block of k information symbols into a block of n channel symbols is called an (n,k) block code. The quantity n is called

the block length, and k is called the dimension of the code. Since the code uses n symbols to send k message symbols, it is said to have code rate k/n.

In convolutional coding, the channel encoder processes the information continuously and associates each long (perhaps semi-infinite) information sequence with a code sequence containing somewhat more digits. The encoder breaks its input sequence into k_0-symbol blocks, where k_0 is usually a small number. These k_0-tuple information symbols are sequentially shifted through a K-stage shift register. Then, on the basis of this k_0-tuple and the preceding (K-1) k_0-tuple information symbols, an n_0-symbol section of the code sequence is generated and transmitted. The length K of the shift register is called the constraint length of the code, and the code rate is k_0/n_0.

Block codes and convolutional codes have similar error correcting capabilities and the same fundamental limitations. In particular, Shannon's fundamental theorem for the discrete noisy channel holds for both types of codes. However, from a practical standpoint, the essential limitation of all coding and decoding

schemes proposed to date has not been Shannon's capacity but the complexity (and cost) of the decoder. For this reason, efforts have been directed toward the design of coding and decoding schemes that could be easily implemented.

1.2 MAJORITY-LOGIC DECODABLE CODES

A significant development in the study of block codes was the introduction of group codes or linear block codes by Slepian [42], [43]. These codes were a generalization of the error-correcting codes of Hamming and all "Systematic Codes" [16]. The codewords of linear block codes were shown to correspond to the elements of a suitably defined group. In general, the code words of an (n,k) linear block code over a finite field GF(q) define a k-dimensional subspace of the space of n-tuples of elements of GF(q). Thus, a linear (n,k) block code contains q^k code vectors.

Even more structure was introduced by the discovery of a subset of the linear block codes called cyclic codes. Cyclic codes are characterized by the

additional property that any cyclic permutation of the digits of a code word is also a code word. Cyclic codes are reasonably practical as they can be encoded and decoded by relatively simple devices called linear feedback shift registers.

Among the classes of cyclic codes, that of majority-logic-decodable codes is an interesting one. This is because the majority-logic decoding scheme possesses the following advantages:

(1) It can be very easily implemented;
(2) the decoding delay is short; and
(3) it can outperform the bounded distance decoding.

For the above reasons, majority-logic decodable cyclic codes are very suitable for error control purposes in high speed digital data transmission systems.

In the class of majority-logic decodable codes, the 1-step decodable codes can be most simply and quickly implemented [27], [41], since they employ a single majority-logic gate. However, since there are only a few codes that are so decodable, there is also

considerable interest in majority-logic decoding with more than one step [12]. A 2-step majority-logic decoder, for instance, will estimate in the first step a set of J parity-check sums which are orthogonal on a particular error digit e_0. Such a set will then be used in the second decoding step to estimate error digit e_0.

Finite geometry codes [8], [10] is a large class of multi-step majority-logic decodable codes. One subclass of these codes is that of the Projective Geometry (PG) codes. This class of cyclic codes contains as proper subclasses the difference-set codes and the maximum-length codes. Also, the even-weighted subset of shortened Reed-Muller (RM) codes [29], [35] fall in this class. The codes are defined as orthogonal on some tactical configurations in finite projective geometries [3], [11], [26].

Just like any linear cyclic code, the code vectors of an (n,k) PG code satisfy a set of n-k linearly independent linear equations. In other words, any vector of the code is orthogonal to the (n-k) x n matrix of the coefficients of these n-k linear equations. This matrix is usually called the parity-

check matrix, and its n-k rows define the parity-check rules, which are the linear equations satisfied by any vector of the code. Obviously, any vector in the (n-k)-dimensional space generated by the rows of the parity-check matrix also defines a parity-check rule. We shall refer to the parity-check space defined by the nonzero vectors of the dual space of the code. This latter is usually called the dual of the PG code.

An (r,s)th-order binary projective geometry code of length $n=(q^{m+1}-1)/(q-1)$, with $q=2^s$, is the largest cyclic code over GF(2) whose dual space (i.e., parity-check space) contains the incidence vectors of all the r-flats in the projective geometry PG(m,q). By an incidence vector of an r-flat of PG(m,q), we mean a binary n-tuple with ones in those positions corresponding to points in the r-flat and zeros elsewhere.

Following the correspondences of the projective geometries PG(m,q) and the Galois field $GF(q^{m+1})$, the conditions characterizing generator polynomials of binary projective geometry codes have been established [15]. From these conditions, one can be better off to

rely on the basic approaches given in the book by Berlekamp [1] or the book by Peterson and Weldon [31] to find roots of the generator polynomials. However, those approaches involve the tabulation of cyclotomic cosets, the evaluation of radix-q weights, and the selection of valid roots power. For classes of long projective geometry codes, the method becomes too cumbersome to be applied. In this context, the author has proposed a method [18] for certain subclasses of binary PG codes.

The most thoroughly studied majority-logic decoding algorithm for projective geometry codes is the step-by-step Reed decoding algorithm. This algorithm was devised in 1954 by Reed [35] for the class of codes discovered by Muller [29]. Reed's algorithm was later extended and generalized by many other coding investigators. Unified formation of majority-logic decoding algorithms can be found in the monograph by Massey [27].

With conventional orthogonal parity-check sums, an (r,s)th-order PG code requires r decoding steps. Improvements for decoding PG codes were suggested by Chen [5] and by Weldon [48]. The concept of

nonorthogonal parity-check sums has also brought out that many more cyclic codes can be decoded with majority-logic decoding, such as the duals of primitive polynomial codes [20], the generalized finite geometry codes [9], [17], and others.

1.3 OUTLINE OF THE THESIS

We now conclude this Introduction with an outline of the thesis.

In chapter II, we develop the background material necessary for the following chapters. First, in Section 2.1, the basic elements of finite field theory are given. This introduces the important concepts of the Galois fields, the cyclotomic cosets, and the minimal functions. In Section 2.2, the factorization of binary primitive polynomials over GF(q), with $q=2^s$, is developed to obtain q-ary irreducible factors. These factors are useful for providing logics on constructing points (or 0-flats) of the projective geometry PG(m,q), m being the dimension of the geometry.

In Section 2.3, the correspondences of the q-ary field elements of $GF(q^{m+1})$ and the points of the projective geometry PG(m,q) are indicated. Important properties of the configurations of PG(m,q) are defined. The (r,s)th-order binary projective geometry codes of length $n=(q^{m+1}-1)/(q-1)$ are introduced in Section 2.4. These codes are defined as the classes of cyclic codes whose parity-check space contains all the incidence vectors of the r-flats in PG(m,q). With this definition, the roots characterizing generator polynomials of binary PG codes are pointed out. Some known results on the dimensions of certain subclasses of PG codes are also indicated.

In chapter III, based on the specified conditions characterizing generator polynomial roots, several results on coset representatives for finding generator polynomials of binary PG codes are developed. First, in Section 3.1, we present an approach for the families of (m-1,s)th- and (m-2,2)th-order binary PG codes. This approach utilizes all possible 2's exponent combinations to form aeceptable integers which characterize roots of the generator polynomial. A coset partition is then made on those valid odd numbers and different cyclotomic

cosets are formed. From each of these cyclotomic cosets, an odd number is selected as the representative.

Unfortunately, the above approach becomes too cumbersome when the values of m and/or s grow large. Concerning this, a slightly different approach on coset representatives is developed in Sections 3.2 and 3.3. This approach is based on the concept that the cyclotomic coset containing a positive integer ν is the union of the cyclic class containing ν and the cyclic classes containing $2^{\ell}\nu$'s, with $0 \leq \ell < s$.

In Section 3.2, on the basis of the above concept of cyclic classes, we derive some sharper results for the families of binary PG codes with parameters m-r=1 and s=1,2. Then in Section 3.3, following the same line of thinking, we tabulate a sharp result for the coset representatives of the binary (r=m-2,s=2)th-order PG codes. The derivations treated here are lengthy and laborious; but the results obtained are useful for quick consultation.

To make clear what we discussed in Sections 3.1 to 3.3, some specific examples are taken in Section 3.4 to

demonstrate the essential ideas of finding coset representatives. We take the (1,3)th-, the (2,2)th-, and the (1,2)th-order binary PG codes with lengths of 73 and 85, respectively, to illustrate our efforts on finding generator polynomials of binary projective geometry codes.

Chapter IV deals with the formation of orthogonal parity-check sets. In Section 4.1, the general concepts of majority-logic decoding and parity-check sets are quoted. The construction of r-subspaces (r-flats) of the projective geometry PG(m,q) is investigated in Section 4.2. First, the points (the 0-flats) of PG(m,q) is generated. This is done through the shift register logics provided by the q-ary primitive polynomial factors developed before on factorizing binary primitive polynomials over GF(q). Then, based on the above 0-flats, a recursive algorithm for constructing r-flats of PG(m,q), r=1, 2,....., m-1, is proposed. The parity-check sums for majority-logic decoding (r,s)th- PG codes are the incidence vectors of these r-flats.

With the above algorithm of constructing r-flats in PG(m,q), it is seen that when the number of decoding

steps and the number of parity-check sums within a check set grow large, the formation of majority-logic gates becomes prohibitive. In this aspect, the so-called sequential-code-reduction (SCR) technique is attractive. This decoding scheme considerably reduces the number of majority gates, but at the cost of increasing the decoding delay.

In Section 4.3, through the so-called generating orthogonal flat f(X), we investigate how the sequential-code-reduction majority-logic decoding technique works. Then, based on the Euclidean algorithm and the Idempotent functions, we give a detailed examination on finding tap-coefficient polynomials $C_i(X)$'s such that at each level of orthogonalization every flat in the check set can be expressed as the product of f(X) and $C_i(X)$, in the sense that the product is taken modulo $1+X^n$.

In Section 4.4, specific examples are used to demonstrate the essential ideas of the formation of parity-check sums. We take the (2,2)th- and the (1,2)th-order binary PG codes of the same length 85 to illustrate our efforts on forming orthogonal parity-check sets.

Finally, in chapter V, conclusions and recommendations for further work are presented based on the results of this study. In addition, we give a partial list, in Appendices A and B, of the generator polynomials and the parity-check sums for a few binary PG codes of moderate lengths.

Chapter II

ALGEBRAIC PRELIMINARIES FOR PROJECTIVE GEOMETRIES AND CODES

The structures of projective geometry codes are connected with the configurations of projective geometries. In this chapter we shall sketch the basic theory for this finite geometry. Certain important properties of the projective geometry PG(m,q) of dimension m and constructed over a finite field GF(q) will be examined.

2.1 GALOIS FIELDS AND MINIMAL FUNCTIONS

A field is a set of elements closed under the operations of 'addition' and 'multiplication', and containing additive and multiplicative identity elements. The addition and multiplication are associative and commutative. The distributive law of multiplication over addition holds in the set. Every field element has a unique additive inverse, and every nonzero element has a unique multiplicative inverse.

The order of a field is the number of elements in the field. If the order is a finite number q, it is called a finite field or a Galois field and is designated by GF(q). Finite fields exist only when the order is a prime number or a power of a prime number, the former are called prime or ground fields, while the latter are called extension fields over the prime fields.

Consider a polynomial P(X) of degree m with coefficients from binary field GF(2)={0,1}:

$$P(X)= p_0 + p_1X + p_2X^2 + \ldots + p_mX^m. \qquad (2.01)$$

If the coefficients p_m of the highest power of X is 1, then polynomial P(X) is called monic. If the polynomial P(X) contains no binary polynomial factors except itself and 1, then P(X) is called irreducible over GF(2). Note that, though a binary irreducible polynomial cannot be factored using only elements from GF(2), it can always have roots in an extension field. Any binary monic polynomial of degree sm can be uniquely factored into monic irreducible polynomials over $GF(2^S)$.

The reciprocal polynomial of P(X) of (2.01), denoted by P*(X), is defined as

$$\begin{aligned} P^*(X) &= X^{\deg P(X)} P(X^{-1}) \\ &= X^m P(X^{-1}) \\ &= p_m + p_{m-1}X + \ldots + p_1 X^{m-1} + p_0 X^m, \end{aligned} \tag{2.02}$$

which is obtained by reversing the order of the coefficients in P(X). It follows that the roots of the reciprocal polynomial P*(X) are the reciprocals of roots of the original polynomial P(X).

A polynomial P(X) is said to have exponent γ if it divides $1+X^{\gamma}$ and not any $1+X^{\delta}$ for any $\delta < \gamma$.

Choose any irreducible polynomial P(X) of degree m and exponent 2^m-1. Let α be a root of P(X)=0. Then the set of all polynomials in α of degree $\leq$ m-1 and coefficients from GF(2), with calculations performed modulo $P(\alpha)$, forms a field of order 2^m. This field is an extension field of GF(2), and is denoted by $GF(2^m)$.

The element α is called a primitive element of the field $GF(2^m)$. In general, any element of $GF(2^m)$ whose

powers generate all the nonzero elements of $GF(2^m)$ is said to be primitive. The irreducible polynomial P(X) of degree m which has a primitive element of $GF(2^m)$ as a root is called a primitive polynomial.

The minimal function or minimal polynomial over GF(2) of element $\alpha^i \in GF(2^m) = \{0,1,\alpha,\alpha^2, \ldots, \alpha^{2^m-2}\}$ is the lowest degree monic polynomial $M_i(X)$ such that $M_i(\alpha^i)=0$. This minimal function possesses the following important properties [21], [26]:

M1) $M_i(X)$ is irreducible;

M2) If f(X) is any polynomial over GF(2) such that $f(\alpha^i)=0$, then $M_i(X) \mid f(X)$;

M3) $M_i(X) \mid X(1+X^{2^m-1})$;

M4) $\deg M_i(X) \leq m$;

M5) The minimal function of a primitive element of $GF(2^m)$ has degree m;

M6) α^i and α^{2i} have the same minimal function.

With respect to property M5), note that if a primitive polynomial P(X) of degree m is used to construct $GF(2^m)$ and $\alpha \in GF(2^m)$ is a root of P(X), then obviously P(X) is the minimal function of α.

On the other hand, by property M6), we can see that the powers of α fall into disjoint sets, which we shall call cyclotomic cosets. All α^j where j runs through a cyclotomic coset have the same minimal function. Basically, a cyclotomic coset containing i is the set

$$C_i = \{i, 2i, 2^2 i, \ldots, 2^{m'-1} i\}, \qquad (2.03)$$

where m' is the smallest positive integer such that $2^{m'} i = i \mod (2^m - 1)$.

Corresponding to each cyclotomic coset C_i of (2.03), the minimal function $M_i(X)$ of $\alpha^i \in GF(2^m)$ can be found by various methods such as Berlekamp's [1]. However, these have been compiled for most cases of practical interest and there are available tables of binary irreducible polynomials like those edited by Peterson and Weldon [31, Appendix C].

In the tables of Peterson and Weldon's, we note that among the given polynomial entries, the first one is always a primitive polynomial. For elements belonging to the same cyclotomic cosets, the corresponding minimal functions are the same, so only

those of coset leaders are listed in the table. Also, of any pair consisting of a polynomial and its reciprocal, only one is listed.

2.2 FACTORS OF BINARY PRIMITIVE POLYNOMIALS OVER GF(2^s) [6]

As we have mentioned, a binary irreducible polynomial cannot be factored over GF(2). However, it can always have roots in an extension field. Let P(X) be a binary primitive polynomial of degree s(m+1). Over GF(q), $q=2^s$, P(X) can be factored as

$$P(X) = P_1(X)P_2(X)\ldots P_s(X), \qquad (2.04)$$

where $P_i(X)$'s are primitive polynomials over GF(q) and each is of degree m+1.

Let $\alpha \in GF(2^{s(m+1)})$ be a primitive root of P(X); then the elements $\alpha, \alpha^{2^1}, \alpha^{2^2}, \ldots, \alpha^{2^{s(m+1)-1}}$ are all of the s(m+1) roots of P(X). These roots are distributed among the polynomial factors $P_i(X)$'s. In general, if $\alpha^{2^{i-1}}$ is a root of the factor $P_i(X)$, $1 \le i \le s$, then the elements $\alpha^{2^{i-1}}, \alpha^{2^{i-1}q}, \alpha^{2^{i-1}q^2}, \ldots, \alpha^{2^{i-1}q^m}$ are among the m+1 roots of $P_i(X)$. Hence

each polynomial factor in (2.04) can be expressed as

$$P_i(X) = (X+\alpha^{2^{i-1}})(X+\alpha^{2^{i-1}}q)\ldots(X+\alpha^{2^{i-1}}q^m). \tag{2.05}$$

To manipulate the above expressions, it is necessary to find out the relationship between the primitive element β of GF(q) and the primitive element α of $GF(q^{m+1})$. From the definition of a primitive element, we see that

$$\beta^{q-1} = 1 \quad \text{and} \quad \alpha^{q^{m+1}-1} = 1, \tag{2.06}$$

hence we have

$$\beta = \alpha^{(q^{m+1}-1)/(q-1)} = \alpha^n. \tag{2.07}$$

After finding the expression of β in terms of α, one simply needs to expand each of the factors of (2.05). By using the logic provided for constructing $GF(q^{m+1})$, one can express all coefficients of $P_i(X)$ into forms of combinations of those basis elements of $GF(q^{m+1})$: α^0, α^1, α^2, ..., α^m. Also, from the relationship of $\beta=\alpha^n$, one can express all elements of GF(q) into some combinations of powers of α. Therefore the coefficients of each polynomial factor $P_i(X)$ can be written in terms of elements from GF(q).

In practice, as long as the first factor $P_1(X)$ is found, it is not necessary to use the same procedures to find $P_2(X)$, $P_3(X)$, ..., $P_s(X)$. As can be seen from (2.05), all roots of $P_2(X)$ are the squares of those of $P_1(X)$, all roots of $P_3(X)$ are the twice squares of those of $P_1(X)$, and so forth. Hence if $P_1(X)$ is of the form

$$P_1(X) = X^{m+1} + p_{1m}X^m + \ldots + p_{11}X + p_{10}, \qquad (2.09)$$

then all coefficients of $P_i(X)$'s will be

$$p_{ij} = (p_{1j})^{2^{i-1}}, \qquad (2.10)$$

where $1 \le i \le s$, $0 \le j \le m$, and $p_{ij} \in GF(q)$. Therefore, once one finds the expression for $P_1(X)$, the other polynomials $P_2(X)$, $P_3(X)$, ..., $P_s(X)$ can all be obtained by simply using the coefficients relationship (2.10).

To clarify the above discussions, let us consider the case of $s(m+1)=6$ and $q=2^s=4$. In this case, the polynomial $P(X)=1+X+X^6$ is a primitive polynomial over GF(2). Hence if α is the primitive root of $1+X+X^6=0$, then the logic $\alpha^6=1+\alpha$ can be used to construct the field $GF(2^6)$. In such a field, every element can be written as a binary combination of the basis elements α^0, α^1, α^2, α^3, α^4, and α^5.

The polynomial $P(X)=1+X+X^6$ can be factored into two primitive polynomials over GF(4), each of degree 3:

$$\begin{aligned} P(X) &= P_1(X).P_2(X) \\ &= (X^3+p_{12}X^2+p_{11}X+p_{10}) \\ &\quad .(X^3+p_{22}X^2+p_{21}X+p_{20}), \end{aligned} \tag{2.11}$$

where $p_{ij} \in GF(4)$, and $p_{2j}=(p_{1j})^2$ for $0 \le j \le 2$.

The roots of P(X) are α, α^2, α^4, α^8, α^{16}, and α^{32}. These roots are distributed to $P_1(X)$ and $P_2(X)$ as follows:

$$\begin{aligned} P_1(X) &= (X+\alpha)(X+\alpha^4)(X+\alpha^{16}), \\ P_2(X) &= (X+\alpha^2)(X+\alpha^8)(X+\alpha^{32}). \end{aligned} \tag{2.12}$$

Taking the first factor $P_1(X)$ and expanding it, we get the expression

$$\begin{aligned} P_1(X) &= X^3+(\alpha+\alpha^4+\alpha^{16})X^2+(\alpha^5+\alpha^{17}+\alpha^{20})X+\alpha^{21} \\ &= X^3+X^2+(\alpha^5+\alpha^4+\alpha^3+\alpha)X+(\alpha^5+\alpha^4+\alpha^3+\alpha+1), \end{aligned} \tag{2.13}$$

where all coefficients have been expressed as combinations of the basis elements of $GF(2^6)$.

The primitive element β of GF(4) is related to the primitive element α of $GF(2^6)$ as $\beta=\alpha^{21}$. Hence

$$\begin{aligned}\beta &= \alpha^{21} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1,\\ \beta^2 &= \alpha^{42} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha,\\ \beta^3 &= \alpha^{63} = 1.\end{aligned} \tag{2.14}$$

From the above expressions, one easily see that

$$\beta^2 = 1 + \beta \tag{2.15}$$

This is the logic used to construct the field GF(4).

Now, by comparing the coefficients of $P_1(X)$ in (2.13) with those of β and β^2 in (2.14), one immediately gets

$$P_1(X) = X^3 + X^2 + \beta^2 X + \beta. \tag{2.16}$$

The coefficients of $P_2(X)$ then are the squared powers of the corresponding coefficients in $P_1(X)$. That is,

$$P_2(X) = X^3 + X^2 + \beta X + \beta^2. \tag{2.17}$$

Note that the powers of β are manipulated over the logic $\beta^2=1+\beta$. The polynomial $P(X)=1+X+X^6$ is therefore factored into

$$P(X) = (X^3+X^2+\beta^2 X+\beta)(X^3+X^2+\beta X+\beta^2), \tag{2.18}$$

where each factor is a primitive polynomial with coefficients from GF(4).

The approach described above holds for any values of m and $q=2^s$. Hence it is possible to list some results rather than going through all the necessary steps. Table 2.1 shows the factors of certain binary primitive polynomials over different Galois fields. It is based on these polynomial factors we develop later the construction of finite projective geometries.

2.3 CONFIGURATIONS OF PROJECTIVE GEOMETRIES

A finite projective geometry consists of a finite set of points and a collection of subsets called lines, which satisfy the following axioms [3], [26]:

(i) The set contains a finite number of points. Every line contains at least 3 points.

(ii) There is a unique line (PQ) passing through any two distinct points P and Q.

(iii) If P, Q, and R are noncollinear points and if a line L contains a point A of the line (PQ) and a point B of the line (QR) but does not contain P, Q, or R, then the line L contains a point C of the line (PR).

Table 2.1 Factors of certain binary primitive polynomials.

$q^{m+1}-1$	$q=2^s$	Binary primitive polynomial P(X)	$\beta=\alpha^n$	logic for GF(q)	Factors of P(X) over GF(q)
2^6-1	2^2	X^6+X+1	$\beta=\alpha^{21}$	$\beta^2=\beta+1$	$(X^3+X^2+\beta^2X+\beta)(X^3+X^2+\beta X+\beta^2)$
2^8-1	2^2	$X^8+X^4+X^3+X^2+1$	$\beta=\alpha^{85}$	$\beta^2=\beta+1$	$(X^4+X^3+\beta X^2+\beta X+\beta)(X^4+X^3+\beta^2X^2+\beta^2X+\beta^2)$
2^9-1	2^3	X^9+X^4+1	$\beta=\alpha^{73}$	$\beta^3=\beta+1$	$(X^3+\beta X^2+\beta^5X+\beta)(X^3+\beta^2X^2+\beta^3X+\beta^2)$ $(X^3+\beta^4X^2+\beta^6X+\beta^4)$
$2^{10}-1$	2^2	$X^{10}+X^3+1$	$\beta=\alpha^{341}$	$\beta^2=\beta+1$	$(X^5+X^4+\beta^2X^3+\beta X^2+\beta)$ $(X^5+X^4+\beta X^3+\beta^2X^2+\beta^2)$
$2^{12}-1$	2^2	$X^{12}+X^6+X^4+X+1$	$\beta=\alpha^{1365}$	$\beta^2=\beta+1$	$(X^6+X^2+X+\beta)(X^6+X^2+X+\beta^2)$
	2^3		$\beta=\alpha^{585}$	$\beta^3=\beta+1$	$(X^4+\beta X^3+\beta^3X^2+X+\beta)$ $(X^4+\beta^2X^3+\beta^6X^2+X+\beta^2)$ $(X^4+\beta^4X^3+\beta^5X^2+X+\beta^4)$
	2^4		$\beta=\alpha^{273}$	$\beta^4=\beta+1$	$(X^3+\beta^{10}X^2+\beta^{13}X+\beta)(X^3+\beta^5X^2+\beta^{11}X+\beta^2)$ $(X^3+\beta^{10}X^2+\beta^7X+\beta^4)(X^3+\beta^5X^2+\beta^{14}X+\beta^8)$
$2^{14}-1$	2^2	$X^{14}+X^{10}+X^6+X+1$	$\beta=\alpha^{5461}$	$\beta^2=\beta+1$	$(X^7+X^4+\beta^2X^3+X^2+\beta^2X+\beta)$ $(X^7+X^4+\beta X^3+X^2+\beta X+\beta^2)$

(iv) For any point P there are at least two lines not containing P, and for any line L there are at least two points not on L.

(v) If P and Q are distinct points of a subspace of the projective geometry, then the subspace contains all points of the line (PQ).

The most important examples of projective geometries are those constructed from the finite field GF(q). Consider the Galois field $GF(q^{m+1})$ which contains GF(q) as a subfield, where q is some power of a prime, say $q=2^s$. Let α be a primitive element in $GF(q^{m+1})$. Then every nonzero element of $GF(q^{m+1})$ can be written uniquely as a power of α, and using the minimal function, we can identify uniquely every such power with an (m+1)-tuple over GF(q). Let $n=(q^{m+1}-1)/(q-1)$; then the order of $\beta=\alpha^n$ is q-1; i.e., α^{q-1} is a primitive nth root of unity. The q elements $0, 1, \beta, \beta^2, \ldots, \beta^{q-2}$ form the Galois field GF(q).

The projective geometry PG(m,q) consists of $(q^{m+1}-1)/(q-1)$ points and is defined by working with the nonzero elements of $GF(q^{m+1})$. There are $q^{m+1}-1$ such nonzero elements, and they are divided into

$(q^{m+1}-1)/(q-1)$ sets, each set constituting one point of PG(m,q). Each point of PG(m,q) has q-1 elements of $GF(q^{m+1})$ mapped onto it, and this projection suggests the name projective geometry.

The rule for forming these sets is that whenever α^i is a nonzero element in $GF(q^{m+1})$ and β^j is any nonzero field element from GF(q), α^i and $\beta^j\alpha^i$ are in the same set; that is, they are mapped onto the same point of PG(m,q). Hence, the points of the projective geometry can be identified with the $n=(q^{m+1}-1)/(q-1)$ distinct one-dimensional linear subspaces of $GF(q^{m+1})$ over GF(q).

Specifically, with α and $\beta=\alpha^n$ being the respective primitive elements of $GF(q^{m+1})$ and GF(q), we partition the nonzero elements of $GF(q^{m+1})$ into the following disjoint subsets:

$$\{\alpha^0, \beta\alpha^0, \beta^2\alpha^0, \ldots, \beta^{q-2}\alpha^0\},$$
$$\{\alpha^1, \beta\alpha^1, \beta^2\alpha^1, \ldots, \beta^{q-2}\alpha^1\},$$
$$\{\alpha^2, \beta\alpha^2, \beta^2\alpha^2, \ldots, \beta^{q-2}\alpha^2\},$$
$$\vdots$$
$$\{\alpha^{n-1}, \beta\alpha^{n-1}, \ldots, \beta^{q-2}\alpha^{n-1}\}. \qquad (2.19)$$

Each set consists of q-1 elements and each element is a multiple of the first element in the set. No element in one set can be a product of an element of GF(q) and an element from a different set. Now, we represent each set by its first element as

$$(\alpha^i) = \{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots, \beta^{q-2}\alpha^i\}, \qquad (2.20)$$

with $0 \le i < n$. Then the n sets (α^0), (α^1), (α^2), ..., (α^{n-1}) will constitute the points of PG(m,q). Note that, for any α^j in $GF(q^{m+1})$, if $\alpha^j = \beta^u\alpha^i$ with $0 \le i < n$, then α^j is represented by (α^i); the q-1 elements in $\{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots, \beta^{q-2}\alpha^i\}$ are considered to be the same point.

The projective geometry PG(m,q) of dimension m over the field GF(q) is the set of the above $n=(q^{m+1}-1)/(q-1)$ points together with collections of subsets called r-flats, for r=0, 1, 2, ..., m-1, which are recursively defined as follows: The 0-flats are the points of PG(m,q). For any r-flat $F^{(r)}$ and any point (α^j) not belonging to $F^{(r)}$, there is a unique (r+1)-flat $F^{(r+1)}$ in PG(m,q) which contains the set of points

$$F^{(r+1)} = \{(\alpha^j)\} \cup \{(\alpha^i + a_2\alpha^j)\}$$
$$= \{(a_1\alpha^i + a_2\alpha^j)\}, \qquad (2.21)$$

where $(\alpha^i) \in F^{(r)}$ and $a_1, a_2 \in GF(q)$ are not both zero.

Let (α^{ℓ_0}), (α^{ℓ_1}), ..., (α^{ℓ_r}) be r+1 linearly independent points in PG(m,q). Then the r-flat containing (α^{ℓ_0}), (α^{ℓ_1}), ..., (α^{ℓ_r}) consists of points of the form $(a_0\alpha^{\ell_0} + a_1\alpha^{\ell_1} + \ldots + a_r\alpha^{\ell_r})$, where $a_i \in GF(q)$ and not all a_i's are zero. There are $q^{r+1}-1$ choices for $a_0, a_1, \ldots, a_r$ ($a_0 = a_1 = \ldots = a_r = 0$ is not allowed). Since there are always q-1 choices of a_0 to a_r resulting in the same point in PG(m,q), there are

$$\begin{bmatrix} r+1 \\ 1 \end{bmatrix} = \frac{q^{r+1}-1}{q-1}$$
$$= q^r + q^{r-1} + \ldots + q + 1 \qquad (2.22)$$

points in an r-flat in PG(m,q).

At this point, it should be noted that, because of the properties of the underlying vector space, any r-flat in PG(m,q) has itself the structure of a projective geometry PG(r,q).

Following the above definition of projective geometry, the number of r-flats in PG(m,q) can be determined. It is the number of ways of choosing r+1 independent points in PG(m,q) divided by the number of ways of choosing r+1 independent points in PG(r,q). That is, the number of PG(r,q) contained in a PG(m,q) is

$$\begin{bmatrix} m+1 \\ r+1 \end{bmatrix} = \prod_{j=0}^{r} \frac{q^{m+1} - q^{j}}{q^{r+1} - q^{j}} \qquad (2.23)$$

This is a quantity known as the q-ary Gaussian binomial coefficient [26]. A similar argument proves that, for any μ and r, with $\mu < r < m$, a given PG(μ,q) is properly contained in exactly

$$\begin{bmatrix} m-\mu \\ r-\mu \end{bmatrix} = \prod_{j=0}^{r-\mu-1} \frac{q^{m-\mu} - q^{j}}{q^{r-\mu} - q^{j}} \qquad (2.24)$$

distinct PG(r,q) which are contained in a given PG(m,q).

Let $(\alpha^{\ell r})$ be a point not in the r-flat

$$F_1^{(r)} = \{(a_0\alpha^{\ell 0} + a_1\alpha^{\ell 1} + \ldots + a_r\alpha^{\ell r})\}.$$

Then the r-flat $F_1^{(r)} = \{(a_0\alpha^{\ell 0} + a_1\alpha^{\ell 1} + \ldots + a_{r-1}\alpha^{\ell r-1} + a_r\alpha^{\ell r})\}$ and the r-flat $F_2^{(r)} = \{(a_0\alpha^{\ell 0}, + a_1\alpha^{\ell 1} + \ldots + a_{r-1}\alpha^{\ell r-1} + a_r\alpha^{\ell r})\}$ have

$$F^{(r-1)} = \{(a_0\alpha^{\ell_0} + a_1\alpha^{\ell_1} + \ldots + a_{r-1}\alpha^{\ell_{r-1}})\}$$

as a common (r-1)-flat. We say that they intersect at $F^{(r-1)}$. The number of r-flats in PG(m,q) that intersect on a given (r-1)-flat in PG(m,q) is

$$\begin{bmatrix} m-r+1 \\ 1 \end{bmatrix} = \frac{q^{m-r+1} - 1}{q - 1}$$

$$= q^{m-r} + q^{m-r-1} + \ldots + q + 1. \qquad (2.25)$$

Every point outside an (r-1)-flat $F^{(r-1)}$ is in one and only one of the r-flats intersecting on $F^{(r-1)}$.

2.4 BINARY PROJECTIVE GEOMETRY (PG) CODES

Associated with each r-flat $F^{(r)}$ of PG(m,q), there is a binary incidence vector $\mathbf{f}$ of length $n=(q^{m+1}-1)/(q-1)$,

$$\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}),$$

whose ith component is

$$f_i = \begin{cases} 1, & \text{if } (\alpha^i) \text{ is a point in } F^{(r)}; \\ 0, & \text{otherwise.} \end{cases}$$

In terms of polynomial, we have

$$f(X) = f_0 + f_1X + f_2X^2 + \ldots + f_{n-1}X^{n-1},$$

which is called the incidence polynomial of the r-flat $F^{(r)}$.

In this thesis, if no confusion arises, the term 'flat' will be used to denote both the point set and the associated incidence vector or incidence polynomial. Also, due to the definition of projective geometry corresponding to the idea of conventional projective space, we will interchangeably call 0-flats, 1-flats and 2-flats points, lines and planes, respectively.

An (r,s)th-order binary projective geometry code of length $n=(q^{m+1}-1)/(q-1)$, with $q=2^s$, is the largest cyclic code over GF(2) whose parity-check space contains the incidence vectors of all the r-flats in PG(m,q).

To characterize the generator polynomial of a binary projective geometry code, it is necessary to define the so-called radix-q weight. Let ν be a nonnegative integer less than $q^{m+1}-1$. This integer can

be expressed in radix-q form as follows:

$$\nu = w_0 + w_1 q + w_2 q^2 + \ldots + w_m q^m, \qquad (2.26)$$

where $0 \le w_i \le q-1$ for $0 \le i \le m$. The radix-q weight of ν is defined as

$$W_q(\nu) = w_0 + w_1 + w_2 + \ldots + w_m, \qquad (2.27)$$

with addition over the integer field.

From Eqs. (2.26) and (2.27), the difference between ν and $W_q(\nu)$ can be expressed as

$$\nu - W_q(\nu) = \sum_{i=1}^{m} w_i(q^i - 1). \qquad (2.28)$$

Since the right side of (2.28) is divisible by q-1, $\nu - W_q(\nu)$ must be divisible by q-1. Hence, if ν is divisible by q-1, then $W_q(\nu)$ is divisible by q-1. Furthermore, it is clear that $2^\ell \nu \bmod q^{m+1}-1$ is divisible by q-1, and hence $W_q(2^\ell \nu)$ is divisible by q-1.

Let V be an (r,s)th-order binary PG code with length $n=(q^{m+1}-1)/(q-1)$, where $q=2^s$. Let g(X) be the generator polynomial of V. With reference to the $GF(q^{m+1}) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{n(q-1)-1}\}$, the element α^ν is a root of g(X) if and only if ν is of the form

$\nu=i(q-1)$ and $\max_{0\le\ell<s} W_q(2^\ell\nu) \le (m-r)(q-1)$, where $W_q(2^\ell\nu)$ is the weight of the radix-q representation of $2^\ell\nu$ [1]. The code V is r-step majority-logic decodable with $J=(q^{m-r+1}-1)/(q-1)$ orthogonal check-sums so that the code has a minimum distance of at least J+1.

Alternatively, the code V can be characterized by the roots of parity-check polynomial $h(X)=(1+X^n)/g(X)$. With α being the primitive element of $GF(q^{m+1})$, the element α^ν is a root of h(X) if and only if ν is of the form $\nu=i(q-1)$ and $\max_{0\le\ell<s} W_q(2^\ell\nu) \ge (m-r+1)(q-1)$, where $W_q(2^\ell\nu)$ is the radix-q weight of the number $2^\ell\nu$.

In the above, we note that all numbers $2^\ell\nu$ are taken modulo $q^{m+1}-1$; that is, modulo n(q-1). We also note that $\nu=0$ is always valid for α^ν to be a root of g(X) and not a root of h(X). That is, 1+X is always a factor of g(X) and not a factor of h(X). Since g(X) and h(X) are dual, finding either of them assures finding the other, we will consider here the generator polynomial only. The same procedure surely can be applied to that of finding parity-check polynomial. Since $\nu=0$ is always valid for characterizing g(X), we will restrict our discussion hereafter to nonzero numbers of ν.

Prange [33] first studied certain projective geometry codes in the late 1950s. He obtained many results on the distribution of weights of the codewords and coset leaders of certain binary codes based on projective geometries and on the permutation groups which preserve these codes.

Weldon [46] introduced difference-set codes, a subclass of the projective geometry codes. The more general classes of codes based on finite geometry were first presented by Rudolph [36], [37]. Further results on projective geometry codes have been obtained by Graham and MacWilliams [14], Goethals and Delsarte [10], MacWilliams and Mann [25], Smith [44], and Hartmann, et al. [17].

Following the above pioneering work, certain results on projective geometry codes have been obtained. For example, the dimension k of the (r,1)th-order binary PG code of length $n=2^{m+1}-1$ is found to be

$$k = \sum_{i=1}^{r} \binom{m+1}{i} \qquad (2.29)$$

These codes are equivalent to the even-weighted subset of the shortened Reed-Muller codes. On the other hand,

the subclass obtained with m+1=3 and r=1 is the class of Weldon's difference-set cyclic codes, for which Graham and MacWilliams [14] proved that

$$n - k = 1 + 3^s. \tag{2.30}$$

For the case of s=1, the (1,1)th-order PG code becomes a maximum-length code which is 1-step majority-logic-decodable.

There is no simple formula for enumerating the number of parity-check bits for general (r,s)th-order PG code. Rather complicated combinatorial expressions for the number of parity-check digits of PG codes can be found in References 15 and 22. However for r=m-1, the number of parity-check digits for the (m-1,s)th-order PG code of length $n = (2^{s(m+1)}-1)/(2^s-1)$ is

$$n - k = 1 + (m+1)^s. \tag{2.31}$$

This expression was obtained independently by Goethals and Delsarte [10], MacWilliams and Mann [25], and Smith [44].

Chapter III

COSET REPRESENTATIVES FOR FINDING GENERATOR POLYNOMIALS

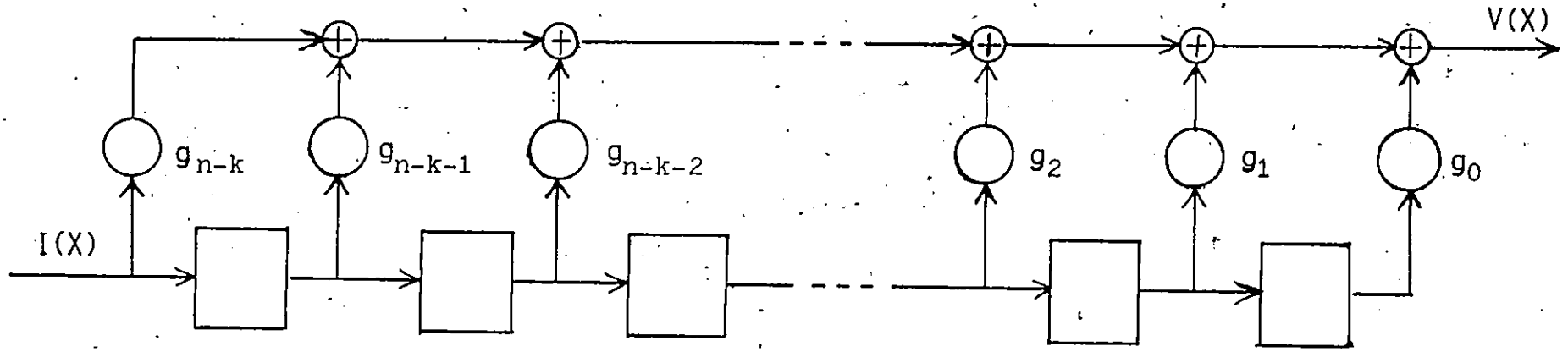
Finding generator polynomial is central to the design of linear feedback shift register encoding circuit of an (n,k) projective geometry code. This family of cyclic codes can be encoded nonsystematically by multiplication of the information polynomial I(X) with the fixed generator polynomial $g(X)= g_0 +g_1X +g_2X^2 +\dots +g_{n-k}X^{n-k}$ to get the codeword V(X). Figure 3.1a shows such a multiply-by-g(X) shift register circuit. With this (n-k)-stage encoding circuit, we simply break the information sequence into k-bit blocks, insert a pad of n-k zeros after every block, and pass the padded stream through the encoder. This provides a sequence of noninteracting n-bit codewords.

It is also possible to encode an (n,k) PG code with a k-stage shift register. This is accomplished by using the parity-check polynomial $h(X)= (1-X^n)/g(X)= h_0 +h_1X +h_2X^2 +\dots +h_kX^k$. A systematic encoder having

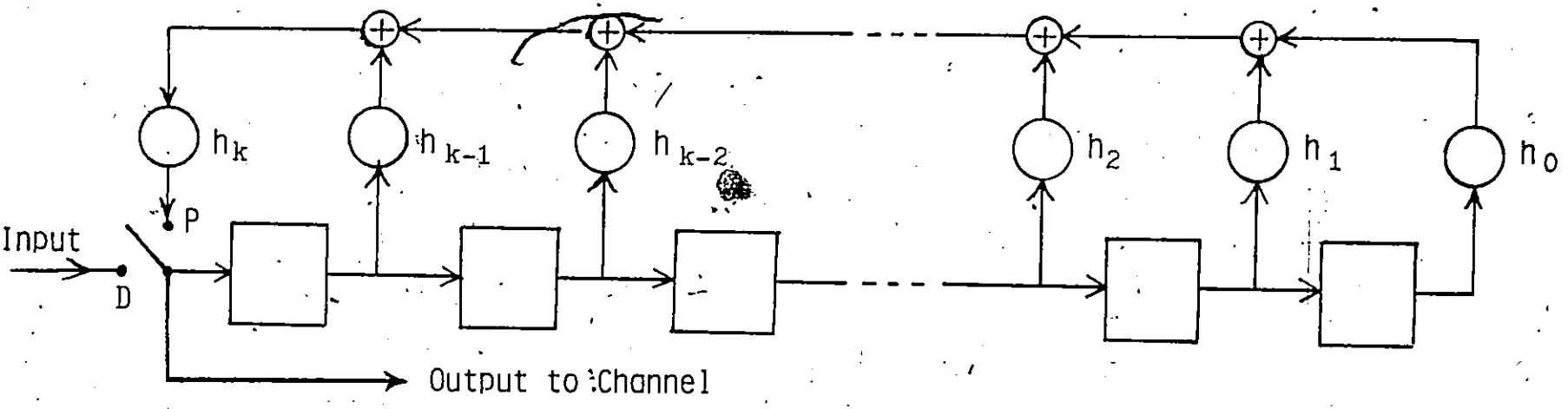
feedback connections based on the above h(X) is shown in Figure 3.1b. With the initial switch position at D, the k information bits are shifted simultaneously into the register and the communication channel. Then the switch is thrown from position D to P, and the register is shifted n-k times. The n-k parity-check bits formed at P are therefore shifted into the channel after the data.

Previously in Section 2.4, we mentioned if α is the primitive element of $GF(q^{m+1})$, with $q=2^s$, then for any (r,s)th-order binary PG code V of length $n=(q^{m+1}-1)/(q-1)$, the element $\alpha^\nu \in GF(q^{m+1})$ is a root of the generator polynomial g(X) if and only if ν is divisible by q-1 and $\max_{0\le \ell<s} W_q(2^\ell \nu) \le (m-r)(q-1)$, where $W_q(2^\ell \nu)$ is the weight of the radix-q representation of the number $2^\ell \nu \bmod n(q-1)$.

Regarding the above, we let S_ν be the set of nonzero numbers $\nu=i(q-1)$ such that α^ν is a root of g(X). Since the code V is binary, we can partition S_ν into cyclotomic cosets. From the conditions characterizing generator roots, we know that we have to examine the numbers q-1, 2(q-1), ..., (n-1)(q-1). Among these numbers, if i(q-1) is an acceptable number for some



(a) (n-k)-stage multiply-by-g(X) circuit, $g(X)=g_0+g_1X+\ldots+g_{n-k}X^{n-k}$.



(b) k-stage systematic encoder based on $h(X)=h_0+h_1X+\ldots+h_kX^k$.

Figure 3.1 Shift register encoding circuits for an (n,k) PG code.

value of i, then so is every number belonging to the cyclotomic coset containing i(q-1). This suggests that if we can find a coset representative from each acceptable cyclotomic coset, then we can obtain g(X) easily, since we can then refer to a table of irreducible factors as given in, say, Peterson and Weldon [31]. By a coset representative we mean an odd number in the cyclotomic coset.

In this chapter we will use the above line of attack to derive certain results which are useful for solving the problem of finding g(X).

3.1 A WAY OF FINDING COSET REPRESENTATIVES

The conventional approach given in [1] or [31] on finding generator polynomials of binary projective geometry codes involves the tabulation of cyclotomic cosets, the evaluation of radix-q weights, and the selection of acceptable cosets. This series of tasks is tedious and not suitable for very long PG codes. In this regard, we propose here an alternative approach on finding coset representatives for certain classes of binary PG codes.

3.1.1 REPRESENTATIVES FOR (m-1,s)TH- PG CODES

These codes have length $n=(q^{m+1}-1)/(q-1)$ and minimum distance $d \geq q+2$, where $q=2^s$, and are (m-1)-step majority-logic-decodable.

Let $Z_{n(q-1)}$ denotes the set of numbers {0, 1, 2, ..., $n(q-1)-1$}. Since $0 \leq \ell < s$ and m-r=1, we must characterize the number ν in $Z_{n(q-1)}$ which satisfies

$$W_q(2^{\ell}\nu) = q-1, \quad 0 \leq \ell < s. \tag{3.01}$$

This can be easily done by expanding ν in binary form to obtain

$$\nu = b_0 + b_1 2 + b_2 2^2 + \ldots + b_{s(m+1)-1} 2^{s(m+1)-1}, \tag{3.02}$$

where $b_i \in Z_2=\{0,1\}$. By grouping the summands in groups of s, we obtain

$$\begin{aligned} \nu = {} & [b_0 + b_1 2 + \ldots + b_{s-1} 2^{s-1}] \\ & + [b_s + b_{s+1} 2 + \ldots + b_{2s-1} 2^{s-1}] 2^s \\ & + \ldots\ldots \\ & + [b_{ms} + b_{ms+1} 2 + \ldots + b_{(m+1)s-1} 2^{s-1}] 2^{ms}, \end{aligned} \tag{3.03a}$$

which is the radix-q expansion of ν. Consequently,

$$W_q(\nu) = \sum_{i=1}^{m+1} \sum_{j=1}^{s} b_{si-j} 2^{s-j}. \qquad (3.03b)$$

Multiplying ν by 2 modulo $q^{m+1}-1 = 2^{s(m+1)}-1$ corresponds to a cyclic shift of the binary coefficient b_i:

$$\begin{aligned} & 2\nu \bmod (2^{s(m+1)}-1) \\ &= [b_{(m+1)s-1} + b_0 2 + b_1 2^2 + \ldots + b_{s-2} 2^{s-1}] \\ &+ [b_{s-1} + b_s 2 + b_{s+1} 2^2 + \ldots + b_{2s-2} 2^{s-1}] 2^s \\ &+ \ldots\ldots \\ &+ [b_{ms-1} + b_{ms} 2 + b_{ms+1} 2^2 + \ldots + b_{(m+1)s-2} 2^{s-1}] 2^{ms}. \end{aligned} \qquad (3.04a)$$

From this, the radix-q weight of the number 2ν is obtained to be

$$W_q(2\nu) = \sum_{i=1}^{m+1} b_{si-1} + \sum_{i=1}^{m+1} \sum_{j=2}^{s} b_{si-j} 2^{s-j+1}. \qquad (3.04b)$$

Comparing (3.03b) and (3.04b), it is apparent that

$$2W_q(\nu) - W_q(2\nu) = (q-1) \sum_{i=1}^{m+1} b_{si-1}. \qquad (3.05)$$

Substituting the radix-q weights $W_q(\nu) = W_q(2\nu) = q-1$ into Equ. (3.05), we have

$$\sum_{i=1}^{m+1} b_{si-1} = 1. \qquad \text{(real sum)}$$

Applying the same argument s times, we obtain

$$\sum_{i=1}^{m+1} b_{si-j} = 1, \quad j=1, 2, \ldots, s. \qquad (3.06)$$

The integers si-j are the integers in $Z_{n(q-1)}$ which are congruent to s-j modulo s. Hence (3.06) says that the integers in $Z_{n(q-1)}$ satisfying condition (3.01) are precisely the numbers

$$2^{i_0} + 2^{i_1} + 2^{i_2} + \ldots + 2^{i_{s-1}}, \qquad (3.07)$$

where $i_\ell \equiv \ell \bmod s$, $i_\ell \in Z_{s(m+1)}$. We have therefore proved the following

THEOREM 3.1:

With $q=2^s$, $n=(q^{m+1}-1)/(q-1)$, we have that the integers ν in $Z_{n(q-1)}$ whose radix-q weight satisfy $W_q(2^\ell \nu)=q-1$, $0 \le \ell < s$, are precisely the numbers

$$\nu = 2^{i_0} + 2^{i_1} + \ldots + 2^{i_{s-1}},$$

where $i_\ell \equiv \ell \bmod s$, and $i_\ell \in Z_{s(m+1)}$.

In connection with Theorem 3.1, we note that the number of acceptable integers is $(m+1)^s$. Hence, by also using the fact that $1+X$ is a factor of $g(X)$, we have straightaway the following

Corollary 3.1:

If $g(X)$ is the generator polynomial of the $(m-1,s)$th-order binary PG code of length $n=(q^{m+1}-1)/(q-1)$, then the degree of $g(X)$ is given by

$$\deg g(X) = (m+1)^s + 1.$$

Example 3.1: Let $s=2$ and $m+1=6$, so that $s(m+1)=12$. The integers in Z_{12} congruent to 0 and to 1 modulo 2 are:

$$0, 2, 4, 6, 8, 10 \equiv 0 \mod 2$$
$$1, 3, 5, 7, 9, 11 \equiv 1 \mod 2.$$

The numbers in $Z_{2^{12}-1}$ satisfying $W_4(\nu)=W_4(2\nu)=3$ are, according to Theorem 3.1,

$$\begin{array}{lll} 2^0+2^1=3, & 2^2+2^1=6, & 2^4+2^1=18, \\ 2^0+2^3=9, & 2^2+2^3=12, & 2^4+2^3=24, \\ 2^0+2^5=33, & 2^2+2^5=36, & 2^4+2^5=48, \\ 2^0+2^7=129, & 2^2+2^7=132, & 2^4+2^7=144, \\ 2^0+2^9=513, & 2^2+2^9=516, & \text{etc.} \\ 2^0+2^{11}=2049, & 2^2+2^{11}=2052, & \ldots\ldots \end{array}$$

Let S_ν be the set of numbers described in Theorem 3.1, i.e., the set of integers $\nu \in Z_{n(q-1)}$ having the property that $W_q(2^\ell \nu)=q-1$, $0 \le \ell < s$. This set S_ν can be partitioned into cyclotomic cosets. Our next task is to find a representative from each of these cosets. Recall that we are interested only in odd numbers. The integers in S_ν are of the form $2^{i_0}+2^{i_1}+\ldots+2^{i_{s-1}}$, where $i_\ell \in Z_{s(m+1)}$ and $i_\ell \equiv \ell \bmod s$. For convenience we shall represent the number $2^{i_0}+2^{i_1}+2^{i_2}+\ldots+2^{i_{s-1}}$ by the set

$$\langle i_0, i_1, i_2, \ldots, i_{s-1} \rangle. \qquad (3.08a)$$

This number will correspond to an odd integer if and only if $i_0=0$. The only odd integers in the coset containing $<0, i_1, i_2, \ldots, i_{s-1}>$ are seen to be

$$<-i_u, i_1-i_u, \ldots, i_{s-1}-i_u>, \qquad (3.08b)$$

where $i_u \in \{0, i_1, i_2, \ldots, i_{s-1}\}$ and the entries are reduced modulo $s(m+1)$.

Example 3.2: With $s=3$ and $m+1=4$, so that $s(m+1)=12$. The integers in Z_{12} congruent to 0, to 1 and to 2 modulo 3 are:

$$0, 3, 6, 9 \equiv 0 \mod 3$$
$$1, 4, 7, 10 \equiv 1 \mod 3$$
$$2, 5, 8, 11 \equiv 2 \mod 3.$$

The exponents of the odd numbers grouped by cosets are

{<0,1,2>, <0, 1,11>, <0,10,11>}
{<0,1,5>, <0, 4,11>, <0, 7, 8>}
{<0,1,8>, <0, 7,11>, <0, 4, 5>}
{<0,4,2>, <0,10, 2>, <0,10, 8>}
{<0,4,8>}
{<0,7,2>, <0, 7, 5>, <0,10, 5>}.

Therefore coset representatives are

$$2^0+2^1+2^2=7 \equiv 2^0+2^{10}+2^{11}=3073$$
$$2^0+2^1+2^5=35 \equiv 2^0+2^7+2^8=385$$
$$2^0+2^1+2^8=259 \equiv 2^0+2^4+2^5=49$$
$$2^0+2^4+2^2=21$$
$$2^0+2^4+2^8=273$$
$$2^0+2^7+2^2=133.$$

3.1.2 REPRESENTATIVES FOR (m-2,2)TH- PG CODES

These codes have length $n=(q^{m+1}-1)/(q-1)=(4^{m+1}-1)/3$ and minimum distance $d \geq [(q^{m-r+1}-1)/(q-1)]+1=22$, and are (m-2)-step majority-logic-decodable.

Since $0 \leq \ell < s=2$ and $m-r=2$, we have to characterize the numbers ν in $Z_{n(q-1)}$ which satisfy

$$W_q(2^\ell \nu) = q-1,\ 2(q-1), \quad \ell=0,1. \qquad (3.09)$$

To do this, we separate the above condition into four sub-conditions:

$$W_q(\nu) = W_q(2\nu) = q-1; \tag{3.09a}$$

$$W_q(\nu) = W_q(2\nu) = 2(q-1); \tag{3.09b}$$

$$W_q(\nu)=q-1,\ W_q(2\nu)=2(q-1); \tag{3.09c}$$

$$W_q(\nu)=2(q-1),\ W_q(2\nu)=q-1. \tag{3.09d}$$

With respect to (3.09a), it is clear that the same arguments of the case of $m-r=1$ can be applied here. Thus the integers in $Z_{n(q-1)}$ satisfying (3.09a) are precisely the numbers

$$2^{i_0} + 2^{i_1}, \tag{3.10}$$

where $i_\ell \equiv \ell \bmod s$, $i_\ell \in Z_{s(m+1)}$.

As regards the condition (3.09b), if we substitute $W_q(\nu)=W_q(2\nu)=2(q-1)$ into (3.05), and follow the radix-q weights of (3.03b) and (3.04b), we get

$$\sum_{i=1}^{m+1} b_{si-j} = 2, \quad j=1,2. \tag{3.11a}$$

This says that the integers in $Z_{n(q-1)}$ satisfying (3.09b) are precisely the numbers

$$2^{i_0} + 2^{i_0'} + 2^{i_1} + 2^{i_1'}, \tag{3.11b}$$

where $i_\ell, i_\ell' \equiv \ell \bmod s$, $i_\ell, i_\ell' \in Z_{s(m+1)}$, and $i_\ell \neq i_\ell'$.

Similarly, for the condition (3.09c) we have

$$\sum_{i=1}^{m+1} b_{si-1}=0, \quad \sum_{i=1}^{m+1} b_{si-2}=3, \tag{3.12a}$$

and for the condition (3.09d) we have

$$\sum_{i=1}^{m+1} b_{si-1}=3, \quad \sum_{i=1}^{m+1} b_{si-2}=0. \tag{3.12b}$$

Hence the integers in $Z_{n(q-1)}$ satisfying (3.09c) are the numbers

$$2^{i_0} + 2^{i_0'} + 2^{i_0''}, \tag{3.13a}$$

and the integers in $Z_{n(q-1)}$ satisfying (3.09d) are the numbers

$$2^{i_1} + 2^{i_1'} + 2^{i_1''}, \tag{3.13b}$$

where $i_\ell, i_\ell', i_\ell'' \equiv \ell \mod s$, $i_\ell, i_\ell', i_\ell'' \in Z_{s(m+1)}$, and $i_\ell \neq i_\ell' \neq i_\ell''$.

Summarizing the above discussions, we get the following

THEOREM 3.2:

With $q=2^s=4$, $n=(q^{m+1}-1)/(q-1)$, and $m-r=2$, we have that the integers ν in $Z_{n(q-1)}$ whose radix-q weight satisfy $W_q(2^\ell \nu)=q-1, 2(q-1)$, for $\ell=0,1$, are precisely

the numbers

$$2^{i_0}+2^{i_1},\quad 2^{i_0}+2^{i_0'}+2^{i_1}+2^{i_1'},$$
$$2^{i_0}+2^{i_0'}+2^{i_0''},\quad 2^{i_1}+2^{i_1'}+2^{i_1''},$$

where $i_\ell, i_\ell', i_\ell'' \equiv \ell \mod s$, $i_\ell, i_\ell', i_\ell'' \in Z_{s(m+1)}$, and $i_\ell \neq i_\ell' \neq i_\ell''$.

Once the integers described in Theorem 3.2 are found, they can be partitioned into cyclotomic cosets. Again we want to find a representative from each of these cosets, and we are interested only in odd numbers. Representing the numbers $2^{i_0}+2^{i_1}$, $2^{i_0}+2^{i_0'}+2^{i_1}+2^{i_1'}$ and $2^{i_0}+2^{i_0'}+2^{i_0''}$ by the sets $\langle i_0, i_1\rangle$, $\langle i_0, i_0', i_1, i_1'\rangle$ and $\langle i_0, i_0', i_0''\rangle$, respectively, we see that the only odd integers in the coset containing $\langle i_0, i_1\rangle$ are

$$\langle i_0-i_u,\ i_1-i_u\rangle,\quad i_u \in \{i_0,\ i_1\}; \tag{3.14a}$$

the only odd integers in the coset containing $\langle i_0, i_0', i_1, i_1'\rangle$ are

$$\langle i_0-i_u,\ i_0'-i_u,\ i_1-i_u,\ i_1'-i_u\rangle,$$
$$i_u \in \{i_0, i_0',\ i_1, i_1'\}; \tag{3.14b}$$

and the only odd integers in the coset containing $\langle i_0, i_0', i_0'' \rangle$ are

$$\langle i_0 - i_u,\ i_0' - i_u,\ i_0'' - i_u \rangle, \quad i_u \in \{i_0,\ i_0',\ i_0''\}. \qquad (3.14c)$$

Note that all of the above entries are reduced modulo $s(m+1)$.

Example 3.3: With $s=2$ and $m+1=5$, so that $s(m+1)=10$, the integers in Z_{10} congruent to 0 and to 1 modulo 2 are:

$$0,\ 2,\ 4,\ 6,\ 8 \equiv 0 \mod 2$$
$$1,\ 3,\ 5,\ 7,\ 9 \equiv 1 \mod 2.$$

The exponents of the odd numbers $2^{i_0}+2^{i_1}$ grouped by cosets are

$$\{\langle 0, 1\rangle, \langle 0, 9\rangle\}$$
$$\{\langle 0, 3\rangle, \langle 0, 7\rangle\}$$
$$\{\langle 0, 5\rangle\}.$$

The exponents of the odd numbers $2^{i_0} + 2^{i_0'} + 2^{i_1} + 2^{i_1'}$ grouped by cosets are

{<0,2,1,3>, <0,8,1,9>, <0,2,1,9>, <0,8,7,9>}
{<0,2,1,5>, <0,8,3,9>, <0,4,1,9>, <0,6,5,7>}
{<0,2,1,7>, <0,8,5,9>, <0,6,1,9>, <0,4,3,5>}

{<0,2,3,5>, <0,8,1,3>, <0,2,7,9>, <0,8,5,7>}
{<0,2,3,7>, <0,8,1,5>, <0,4,7,9>, <0,6,3,5>}
{<0,2,5,7>, <0,8,3,5>}

{<0,4,1,3>, <0,6,7,9>, <0,2,3,9>, <0,8,1,7>}
{<0,4,1,5>, <0,6,1,7>, <0,4,3,9>, <0,6,5,9>}
{<0,4,1,7>, <0,6,3,7>, <0,6,3,9>, <0,4,3,7>}

{<0,6,1,3>, <0,4,5,7>, <0,2,5,9>, <0,8,3,7>}
{<0,6,1,5>, <0,4,5,9>}.

The exponents of the odd numbers $2^{i_0} + 2^{i'_0} + 2^{i''_0}$ grouped by cosets are

{<0,2,4>, <0,2,8>, <0,6,8>}
{<0,2,6>, <0,4,8>, <0,4,6>}.

From each of the above cyclotomic cosets, one of the odd numbers can be chosen as representative. Converted back into decimals, the representatives are calculated as the 2's power sums of the associated 2's exponent entries.

3.2 SOME SHARPER RESULTS FOR CODES WITH m-r=1 AND s=2,3

Though the procedures discussed in Subsection 3.1.1 prove effective for finding coset representatives, they are generally laborious when s or m becomes large. Concerning this, we have worked on two specific cases (s=2 and s=3) to formulate sharper results. The main ideas behind these results are as follows.

The radix-q representation of a positive integer $2^{\ell}\nu$, $0 \leq \ell < s$, is of the form

$$2^{\ell}\nu = w_0^{(\ell)} + w_1^{(\ell)}q + \ldots + w_m^{(\ell)}q^m, \quad w_i^{(\ell)} \in Z_q.$$

Since a "cyclic shift" of $2^{\ell}\nu$ is the number

$$q.2^{\ell}\nu = w_m^{(\ell)} + w_0^{(\ell)}q + w_1^{(\ell)}q^2 + \ldots + w_{m-1}^{(\ell)}q^m,$$

we conclude that the cyclotomic coset containing a positive integer ν is the union of the cyclic class containing ν and the cyclic classes containing 2ν, 4ν, 8ν, ..., and $2^{s-1}\nu$. Note that by the cyclic class containing $2^{\ell}\nu$, we mean set of $2^{\ell}\nu$ and all of its cyclic shifts.

3.2.1 CODES WITH s=2 AND m-r=1

The codes with s=2 and m-r=1 are of length $n=(4^{m+1}-1)/3$ and minimum distance $d \geq [(4^2-1)/3]+1=6$, and are (m-1)-step majority-logic-decodable.

Since $0 \leq \ell < s=2$ and m-r=1, we are interested in the numbers $\nu=i(q-1)$ and $2\nu=2i(q-1)$ such that $W_4(\nu)=W_4(2\nu)=3$. Regarding this, we have to investigate only the numbers of the forms:

$$\begin{aligned} \nu_{11} &= 1.4^0 + 1.4^i + 1.4^j, \ 1 \leq i < j \leq m; \\ \nu_{12} &= 1.4^0 + 2.4^i, \ 1 \leq i \leq m; \\ \nu_{13} &= 3.4^0. \end{aligned} \tag{3.15}$$

Since $W_4(2\nu_{11})=6$, the coset containing ν_{11} are invalid. Thus we have to deal with only ν_{12} and ν_{13} further. Again, we see that the number ν_{13} is contained in the coset generated by $\nu_{12,2} = 1.4^0+2.4^m$. Thus we are left with only $\nu_{12,1} = 1.4^0+2.4^i$, with $1 \leq i \leq m-1$. We see that $W_4(\nu_{12,1})=3$ and that the coset generated by $\nu_{12,1}$ has in it another number $\nu'_{12,1} = 1.4^0+2.4^{m-i}$, which has the same form as $\nu_{12,1}$ and is a cyclic shift of $2\nu_{12,1}$. This means that, if we choose $i \leq [m/2]$, then each $L_{11}=1+2.4^i$ will produce a different coset.

THEOREM 3.3:

For the case of s=2 and m-r=1, each number

$$L_{11}=1+2.4^{i}, \quad 0 \leq i \leq [m/2],$$

produces a different cyclotomic coset, and the union of these cosets is the set S_{ν}.

In connection with Theorem 3.3, we recall that S_{ν} is the set of nonzero numbers $\nu=i(q-1)$ such that α^{ν} is a root of $g(X)$.

From Theorem 3.3, and referring to the fact that every binary PG code has $1+X$ as a factor of $g(X)$, we straightaway have the following

Corollary 3.3:

For the case of s=2 and m-r=1, the generator polynomial $g(X)$ of the (r,s)th-order binary PG code is given by

$$g(X) = (1+X).G_0(X).G_1(X).....G_{\lambda}(X),$$

where $G_i(X)$ is the minimal function of $\alpha^{1+2.4^{i}} \in GF(2^{s(m+1)})$ and $\lambda=[m/2]$.

Example 3.4: For s=2 and m-r=1, consider the case of m+1=6, so that the code V has $n=(4^6-1)/3=1365$. From Theorem 3.3, we have the coset representatives to be

$$1+2.4^0=3,\ 1+2.4^1=9,\ \text{and}\ 1+2.4^2=33.$$

We see, on referring to the table of irreducible factors in Peterson and Weldon, under the entry of degree s(m+1)=12, that the minimal function of α^3 is $M_3(X)= 1+X+X^3+X^4+X^6+X^{10}+X^{12}$, that of α^9 is $M_9(X)= 1+X^2+X^4+X^5+X^6+X^7+X^8+X^9+X^{12}$, and that of α^{33} is $M_{33}(X)= 1+X^3+X^6+X^7+X^9+X^{10}+X^{12}$. This means, in view of Corollary 3.3, that

$$\begin{aligned} g(X) &= (1+X).M_3(X).M_9(X).M_{33}(X) \\ &= (1+X)(1+X+X^3+X^4+X^6+X^{10}+X^{12}) \\ &\quad (1+X^2+X^4+X^5+X^6+X^7+X^8+X^9+X^{12}) \\ &\quad (1+X^3+X^6+X^7+X^9+X^{10}+X^{12}), \end{aligned}$$

implying that the code V has n-k=37.

3.2.2 CODES WITH s=3 AND m-r=1

For the case of s=3 and m-r=1, we have to investigate the numbers $\nu=i(q-1)$, $2\nu=2i(q-1)$, and

$4\nu=4i(q-1)$ such that $W_8(\nu)= W_8(2\nu)= W_8(4\nu)= 7$. With this requirement of radix-q weight, we need to investigate only the numbers of the following valid forms:

$$\begin{aligned}
\nu_{21} &= 1.8^0 + 6.8^i, \\
\nu_{22} &= 1.8^0 + 2.8^i + 4.8^j, \\
\nu_{23} &= 3.8^0 + 4.8^i, \\
\nu_{24} &= 7.8^0, \\
\nu_{25} &= 5.8^0 + 2.8^i, \\
\nu_{26} &= 1.8^0 + 4.8^i + 2.8^j.
\end{aligned} \tag{3.16}$$

Examining the numbers $\nu_{21}=1+6.8^i$, we have

$$\begin{aligned}
2\nu_{21,1} &= 2 + 4.8^i + 1.8^{i+1}, \text{ if } i<m \\
2\nu_{21,2} &= 3 + 4.8^i, \text{ if } i=m
\end{aligned} \tag{3.17a}$$

and

$$\begin{aligned}
4\nu_{21,1} &= 4 + 3.8^{i+1}, \text{ if } i<m \\
4\nu_{21,2} &= 7, \text{ if } i=m.
\end{aligned} \tag{3.17b}$$

A cyclic shift of $2\nu_{21,1}$ gives

$$\nu'_{21,1} = 1 + 2.8^{m-i} + 4.8^m, \quad i<m, \tag{3.18a}$$

and a cyclic shift of $4\nu_{21,1}$ gives

$$\nu''_{21,1} = 3 + 4.8^{m-i}, \quad i<m. \qquad (3.18b)$$

Comparing $2\nu_{21,2}$, $4\nu_{21,2}$, $\nu'_{21,1}$ and $\nu''_{21,1}$, we easily obtain

LEMMA 3.4.01:

Each number $L_{21}=1+6.8^{i}$, with $1 \le i \le m$, produces a different cyclotomic coset. All numbers of the forms ν_{21}, ν_{22} with $j=m$, ν_{23} and ν_{24} are contained in these cyclotomic cosets.

As for numbers of the form $\nu_{22}=1+2.8^{i}+4.8^{j}$, we have the following

LEMMA 3.4.02:

Each number $L_{22}=1+2.8^{i}+4.8^{j}$, with $1 \le i \le [(m-1)/3]$ and $2i+1 \le j \le m-i$, generates a different cyclotomic coset. In addition, when $m=0 \bmod 3$, the number $L'_{22}=1+2.8^{m/3}+4.8^{2m/3}$ generates another cyclotomic coset. All numbers of the form ν_{22} with $j\neq m$ are contained in these cyclotomic cosets.

Proof:

For the numbers of the form $\nu_{22} = 1+2.8^{i}+4.8^{j}$, we note that the numbers $\nu_{22,2} = 1+2.8^{i}+4.8^{m}$ have already been covered in the cosets of Lemma 3.4.01. Hence we will restrict ourselves here to the numbers $\nu_{22,1} = 1+2.8^{i}+4.8^{j}$, with $j \neq m$. We have

$$\begin{aligned} \nu_{22,1} &= 1 + 2.8^{i} + 4.8^{j}, \; j \neq m; \\ 2\nu_{22,1} &= 2 + 4.8^{i} + 1.8^{j+1}; \\ 4\nu_{22,1} &= 4 + 1.8^{i+1} + 2.8^{j+1}, \end{aligned} \tag{3.19}$$

Now let us consider numbers $L_{22}=1+2.8^{i}+4.8^{j}$, with $2i+1 \leq j \leq m-i$. Since $2i+1$ has to be $\leq m-i$, it immediately follows that $1 \leq i \leq [(m-1)/3]$. The numbers of the form ν_{22}, in the coset of L_{22}, are given by

$$\begin{aligned} L_{22} &= 1 + 2.8^{i} + 4.8^{j}, \\ \nu'_{22,1} &= 1 + 2.8^{m-j} + 4.8^{m-j+i}, \\ \nu''_{22,1} &= 1 + 2.8^{j-i} + 4.8^{m-i}, \end{aligned} \tag{3.20}$$

where $\nu'_{22,1}$ is a cyclic shift of $2\nu_{22,1}$ and $\nu''_{22,1}$ is a cyclic shift of $4\nu_{22,1}$.

Now we ask the question whether or not any, or both, of $\nu'_{22,1}$ and $\nu''_{22,1}$ can belong to $\{L_{22}\}$. Suppose $\nu'_{22,1}$ belongs. Then this means that $2(m-j)+1 \leq m-j+i$

$\leq j$. From the leftmost inequality we have $m-j+1 \leq i$ or $j \geq m-i+1$. This is against the supposition of $j \leq m-i$. Therefore $\nu'_{22,1}$ cannot belong to $\{L_{22}\}$. Next we suppose $\nu''_{22,1}$ belongs. Then this means that $2(j-i)+1 \leq m-i \leq m-(j-i)$. From the rightmost inequality we have $m-i \leq m-j+i$ or $j \leq 2i$. This is against the supposition of $j \geq 2i+1$. Hence $\nu''_{22,1}$ cannot belong to $\{L_{22}\}$. The number $L_{22}= 1+2.8^{i}+4.8^{j}$, with $1 \leq i \leq [(m-1)/3]$ and $2i+1 \leq j \leq m-i$, thus generates a different coset. Each of these cosets contains three numbers of the form ν_{22}.

At this point we have to calculate the number of numbers ν_{22} covered by the cyclotomic cosets generated by L_{22}'s. Setting $m-1=3M+\lambda$, $\lambda \in \{0,1,2\}$, we see, with respect to L_{22}, that the number N_c of the cyclotomic cosets is given by

$$\begin{aligned} N_c &= \Sigma_{i=1}^{M} (m-3i) \\ &= mM - 3M(M+1)/2 \\ &= m\,\frac{m-\lambda-1}{3} - \frac{3}{2}\,\frac{m-\lambda-1}{3}\cdot\frac{m-\lambda+2}{3} \\ &= (m-\lambda-1)(m+\lambda-2)/6. \end{aligned}$$

For $\lambda=0,1$, that is, for $m=1,2 \bmod 3$, N_c reduces to $C(m-1,2)/3$. Here $C(x,y)$ refers to the y-combinations of

x objects and is defined by $C(x,y)=x!/y!(x-y)!$. Since each coset has three numbers of the form $\nu_{22,1}$ and there are altogether $C(m-1,2)$ numbers of the form $\nu_{22,1}$, we conclude that all numbers of the form $\nu_{22,1}$ are covered by the cyclotomic cosets generated by L_{22}'s when $m=1,2 \bmod 3$.

For the case of $\lambda=2$ or $m=0 \bmod 3$, the number N_c becomes $[C(m-1,2)-1]/3$, implying that $C(m-1,2)-1$ numbers of the form $\nu_{22,1}$ are covered in the cosets generated by L_{22}'s. This means that, when m is a multiple of 3, there is one more coset not covered by $\{L_{22}\}$. That coset is the one generated by $L'_{22}=1+2.8^{m/3}+4.8^{2m/3}$. This coset has only one number of the form $\nu_{22,1}$. Thus we have proved the Lemma 3.4.02.

As regards the numbers of the forms $\nu_{25}=5+2.8^{i}$ and $\nu_{26}=1+4.8^{i}+2.8^{j}$, we have the following lemmas.

LEMMA 3.4.03:

Each number $L_{25}=5+2.8^{i}$, with $1 \leq i \leq m-1$, generates a different cyclotomic coset. All numbers of the forms ν_{25} and ν_{26}, with $j=m$ or $j=i+1$, are contained in these cyclotomic cosets.

Proof:

For the numbers of the form $\nu_{25}=5+2.8^{i}$, we have the numbers $2\nu_{25}= 2+1.8^{1}+4.8^{i}$ and $4\nu_{25}= 4+2.8^{1}+1.8^{i+1}$. A cyclic shift of $2\nu_{25}$ gives $\nu'_{25}=1+4.8^{m+i}+2.8^{m}$, and a cyclic shift of $4\nu_{25}$ gives $\nu''_{25}=1+4.8^{m-i}+2.8^{m-i+1}$. Therefore, if we let $L_{25}=5+2.8^{i}$, $1 \le i \le m-1$, then all numbers of the forms ν_{25}, $\nu_{26,1}= 1+4.8^{i}+2.8^{m}$, and $\nu_{26,2}= 1+4.8^{i}+2.8^{i+1}$ are contained in the cyclotomic cosets produced by L_{25}'s. Thus we obtain the stated result.

LEMMA 3.4.04:

Each number $L_{26}=1+4.8^{i}+2.8^{j}$, with $1 \le i \le [(m-2)/3]$ and $2i+1 \le j \le m-i-1$, produces a different cyclotomic coset. In addition, when $m=1 \bmod 3$, the number $L'_{26}=1+4.8^{(m-1)/3}+2.8^{2(m-1)/3+1}$ generates another cyclotomic coset. All numbers of the form ν_{26}, with $j\neq m$ or $j\neq i+1$, are contained in these cyclotomic cosets.

Proof:

For the numbers of the form $\nu_{26}=1+4.8^{i}+2.8^{j}$, we have $2\nu_{26}=2+1.8^{i+1}+4.8^{j}$ and $4\nu_{26}=4+2.8^{i+1}+1.8^{j+1}$. Now let us consider numbers $L_{26}=1+4.8^{i}+2.8^{j}$, with $2i+1 \le j \le m-i-1$. Since $2i+1$ has to be $\le m-i-1$, it

immediately follows that $1 \le i \le [(m-2)/3]$. The numbers of the form ν_{26} in the coset of L_{26} are given by

$$\begin{aligned} L_{26} &= 1 + 4.8^{i} + 2.8^{j}, \\ \nu'_{26} &= 1 + 4.8^{j-i-1} + 2.8^{m-i}, \\ \nu''_{26} &= 1 + 4.8^{m-j} + 2.8^{m-j+i+1}, \end{aligned} \tag{3.21}$$

where ν'_{26} and ν''_{26} are respectively the cyclic shifts of $2\nu_{26}$ and $4\nu_{26}$.

Now we ask the question whether or not ν'_{26} and/or ν''_{26} can belong to $\{L_{26}\}$. Suppose ν'_{26} belongs. Then we get $m-i \le m-i-1$. But this is impossible. Next suppose ν''_{26} belongs. Then this means that $2(m-j)+1 \le m-j+i+1 \le m-i-1$. From this we have $m-j \le i$ or $j \ge m-i$. This contradicts the supposition of $j \le m-i-1$. Hence both ν'_{26} and ν''_{26} cannot belong to $\{L_{26}\}$. The number $L_{26}= 1+4.8^{i}+2.8^{j}$, with $1 \le i \le [(m-2)/3]$ and $2i+1 \le j \le m-i-1$, thus generates a different coset. Each of these cosets contains three numbers of the form ν_{26}.

In calculating the number of numbers ν_{26} covered by the above cyclotomic cosets, let us set $m-2=3M+\lambda$, $\lambda \in \{0,1,2\}$. We see, with respect to L_{26}, that the number N_c of the cyclotomic cosets is given by

$$N_c = \sum_{i=1}^{M} (m-3i-1)$$
$$= (m-1)M - 3M(M+1)/2$$
$$= (m-1)\frac{m-\lambda-2}{3} - \frac{3}{2}\frac{m-\lambda-2}{3}\cdot\frac{m-\lambda+1}{3}$$
$$= (m-\lambda-2)(m+\lambda-3)/6.$$

For $\lambda=0,1$, N_c reduces to $C(m-2,2)/3$, while for $\lambda=2$, N_c reduces to $[C(m-2,2)-1]/3$. Since each coset has three numbers of the form ν_{26}, we have that there are $C(m-2,2)$ numbers of the form ν_{26} covered in the cosets generated by L_{26}'s for $\lambda=0,1$ or $m=0,2 \bmod 3$. If $\lambda=2$ or $m=1 \bmod 3$, there are $C(m-2,2)-1$ numbers of the form ν_{26} covered in the above cyclotomic cosets. Besides, the number of numbers of the form ν_{26} contained in the cosets generated by L_{25}'s is $(m-1)+(m-2)=2m-3$. Therefore the number of numbers of the form ν_{26} contained in the cosets generated by L_{25} and L_{26} is $(2m-3)+C(m-2,2) = C(m,2)$ for $m=0,2 \bmod 3$, and $(2m-3)+C(m-2,2)-1 = C(m,2)-1$ for $m=1 \bmod 3$. Since $C(m,2)$ is the total number of numbers ν_{26}'s, we conclude that, when $m=1 \bmod 3$, there is one more coset not covered by $\{L_{25}\}$ and $\{L_{26}\}$. In searching this coset, we find that $\nu_{26}=\nu'_{26}=\nu''_{26}$ when $i=(m-1)/3$ and $j=2(m-1)/3+1$. This means that the number $L'_{26}=1+4.8^{(m-1)/3}+2.8^{2(m-1)/3+1}$ will

produce another cyclotomic coset. This coset contains only one number of the form ν_{26}. Thus we have proved the Lemma 3.4.04.

On the basis of Lemmas 3.4.01 to 3.4.04, we conclude with the following theorem.

THEOREM 3.4:

For the case of s=3 and m-r=1, the cyclotomic cosets produced by

$$L_{21} = 1+6.8^{i}, \text{ with } 1 \le i \le m,$$

$$L_{22} = 1+2.8^{i}+4.8^{j}, \text{ with } 2i+1 \le j \le m-i,$$

$$L'_{22} = 1+2.8^{m/3}+4.8^{2m/3}, \text{ with } m=3M, M=1,2,\ldots,$$

$$L_{25} = 5+2.8^{i}, \text{ with } 1 \le i \le m-1,$$

$$L_{26} = 1+4.8^{i}+2.8^{j}, \text{ with } 2i+1 \le j \le m-i-1,$$

and

$$L'_{26} = 1+4.8^{(m-1)/3}+2.8^{2(m-1)/3+1},$$

with m=4+3M, M=0,1,2,...,

are disjoint, and the union of these cosets is the set S_{ν}.

Example 3.5: For s=3 and m-r=1, consider the case of m+1=4, so that the code V has $n=(8^4-1)/7=585$. From Theorem 3.4, we have coset representatives to be

$$1+6.8^1=49, \quad 1+6.8^2=385,$$
$$1+6.8^3=3073, \quad 1+2.8^1+4.8^2=273,$$
$$5+2.8^1=21, \quad 5+2.8^2=133.$$

Note that there are no coset representatives corresponding to L_{22}, L_{26} and L'_{26}. Thus, if $M_i(X)$ is the minimal function of α^i, i=21, 49, 133, 273, 385 and 3073, and $\alpha^i \in GF(2^{12})$, then the generator polynomial of V will be

$$g(X) = (1+X).M_{21}(X).M_{49}(X).M_{133}(X) .M_{273}(X).M_{385}(X).M_{3073}(X).$$

3.3 A SHARP RESULT FOR (m-2,2)TH-ORDER BINARY PG CODES

The procedures depicted in Subsection 3.1.2 for finding cyclotomic coset representatives of the (m-2,2)th-order binary PG codes are also cumbersome when the value of m becomes large. Regarding this, we formulate in this section an exact result applicable to any value of m.

For the case of $0 \le \ell < s=2$ and $m-r=2$, we have to investigate the numbers $\nu=i(q-1)$ and $2\nu=2i(q-1)$ such that $W_4(\nu)=3,6$ and $W_4(2\nu)=3,6$. Recall that we are interested only in odd numbers.

With respect to the radix-4 weight $W_4(2^{\ell}\nu)=3$, the possible odd numbers are of the forms

$$\nu_{31} = 1.4^0 + 1.4^i + 1.4^j,\ 1 \le i < j \le m;$$
$$\nu_{32} = 1.4^0 + 2.4^i,\ 1 \le i \le m;$$
$$\nu_{33} = 3.4^0. \qquad (3.22)$$

With reference to the previous discussion on ν_{12} and ν_{13}, a little reflection shows that the same arguments can be applied to ν_{32} and ν_{33}. Thus we have straightaway the following

LEMMA 3.5.01:

Each number $L_{32}=1+2.4^i$, with $0 \le i \le [m/2]$, produces a different cyclotomic coset. All numbers of the forms ν_{32} and ν_{33} are contained in these cosets.

As regards the numbers ν_{31}, we have the following

LEMMA 3.5.02:

Each number $L_{31}=1+4^i+4^j$, with $2i \le j \le m-i$, generates a different cyclotomic coset. In addition, when $m=2 \mod 3$, the number $L'_{31}=1+4^{(m+1)/3}+4^{2(m+1)/3}$ generates another cyclotomic coset. All numbers of the form ν_{31} are contained in these cosets.

Proof:

The numbers of the form ν_{31}, in the coset of L_{31}, are given by

$$\begin{aligned} L_{31} &= 1 + 1.4^i + 1.4^j, \\ \nu'_{31} &= 1 + 1.4^{m+1-j} + 1.4^{m+1-j+i}, \\ \nu''_{31} &= 1 + 1.4^{j-i} + 1.4^{m+1-i}, \end{aligned} \qquad (3.23)$$

which are all cyclic shifts of one another.

Now we ask the question whether or not ν'_{31} and/or ν''_{31} can belong to $\{L_{31}\}$. Suppose ν'_{31} belongs. Then this means that $2(m+1-j) \le (m+1-j+i) \le j-1$. From this we have $j \ge m+1-i$. This contradicts the supposition of $j \le m-i$. Next we suppose ν''_{31} belongs. Then this means that $2(j-i) \le (m+1-i) \le m-(j-i)$. From this we have $j \le 2i-1$. This contradicts the supposition of $j \ge 2i$. Hence the numbers ν'_{31} and ν''_{31} cannot belong to $\{L_{31}\}$. The number $L_{31}=1+4^i+4^j$, with $2i \le j \le m-i$, thus

generates a different cyclotomic coset. Each of these cosets contains three numbers of the form ν_{31}.

In calculating the number of numbers ν_{31} covered by the above cosets, we find that, when m=0,1 mod 3, all numbers of the form ν_{31} are covered by the above cyclotomic cosets; when m=2 mod 3, there is one more coset not covered by $\{L_{31}\}$. That coset is the one generated by $L'_{31} = 1 + 4^{(m+1)/3} + 4^{2(m+1)/3}$. This coset has only one number of the form ν_{31}. Thus we have proved the Lemma 3.5.02.

This concludes the discussion of the numbers $2^{\ell}\nu$, ℓ=0,1, for which $W_4(2^{\ell}\nu)=3$.

Next we consider the case of $W_4(2^{\ell}\nu)=6$. With this radix-4 weight we will look into the odd numbers of the following forms:

$$\begin{aligned}
\nu_{41} &= 1.4^0 + 2.4^i + 3.4^j \\
\nu_{42} &= 1.4^0 + 3.4^i + 2.4^j \\
\nu_{43} &= 3.4^0 + 3.4^i \\
\nu_{44} &= 1.4^0 + 1.4^i + 2.4^j + 2.4^u \\
\nu_{45} &= 1.4^0 + 2.4^i + 1.4^j + 2.4^u. \qquad (3.24)
\end{aligned}$$

Surely, there are other forms of odd numbers having weight $W_4(2^{\ell}\nu)=6$, like the numbers $3+1.4^i+2.4^j$, $3+2.4^i+1.4^j$, and $1+2.4^i+2.4^j+1.4^u$. But, these numbers are respectively the cyclic shifts of ν_{41}, ν_{42}, and ν_{44}; they are contained in the cyclotomic cosets with coset representatives derived from the above listed numbers of (3.24).

Let us start with the numbers $\nu_{41}= 1 +2.4^i +3.4^j$. We have

$$\begin{aligned}
2\nu_{41,1} &= 2+1.4^{i+1}+2.4+1.4^{j+1}, && \text{if } j\neq m,\ i+1\neq j;\\
2\nu_{41,2} &= 3 + 1.4^{i+1} + 2.4^{m}, && \text{if } j=m,\ i+1\neq j;\\
2\nu_{41,3} &= 2 + 3.4^{j} + 1.4^{j+1}, && \text{if } j\neq m,\ i+1=j;\\
2\nu_{41,4} &= 3 + 3.4^{m}, && \text{if } j=m,\ i+1=j.
\end{aligned} \tag{3.25}$$

This means that, if $j\neq m$ and $i+1\neq j$ on the one hand, or if $j=m$ and $i=m-1$ on the other, then the only number, of the form ν_{41}, in the coset of ν_{41}, is ν_{41} itself. This leads that each number $L_{41,1}= 1+2.4^i+3.4^j$, with $j\neq m$ and $i+1\neq j$, or with $j=m$ and $i=m-1$, generates a different coset. In these cosets there is no other number of the form ν_{41}.

Now let us discuss further the cases other than $j\neq m$ and $i+1\neq j$, and $j=m$ and $i+1=j$. For the case of $j=m$ and $i+1\neq j$, we have

$$\nu_{41,2} = 1 + 2.4^{i} + 3.4^{m},$$
$$2\nu_{41,2} = 3 + 1.4^{i+1} + 2.4^{m}. \qquad (3.26a)$$

A cyclic shift of $2\nu_{41,2}$ gives

$$\nu'_{41,2} = 1 + 2.4^{m-i-1} + 3.4^{m-i}. \qquad (3.26b)$$

For the case of $j\neq m$ and $i+1=j$, we have

$$\nu_{41,3} = 1 + 2.4^{i} + 3.4^{i+1},$$
$$2\nu_{41,3} = 2 + 3.4^{i+1} + 1.4^{i+2}. \qquad (3.27a)$$

A cyclic shift of $2\nu_{41,3}$ gives

$$\nu'_{41,3} = 1 + 2.4^{m-i-1} + 3.4^{m}. \qquad (3.27b)$$

Comparing $\nu_{41,2}$ with $\nu'_{41,3}$ and comparing $\nu'_{41,2}$ with $\nu_{41,3}$, we easily obtain that each number $L_{41,2}=1+2.4^{i}+3.4^{m}$, with $1 \leq i \leq m-2$, generates a different coset. Each of them has two numbers of the form ν_{41}.

The number of numbers ν_{41} contained in the cosets generated by $L_{41,1}$ is given by $1+[(m-3)+(m-4)+\ldots+1] = (m^2-5m+8)/2$. The number of numbers ν_{41} contained in

the cosets generated by $L_{41,2}$ is given by $2(m-2)=2m-4$. Therefore the number of numbers ν_{41} contained in the above cosets is given by $(m^2-5m+8)/2 + (2m-4) = C(m,2)$, which is the number of numbers ν_{41}.

Summarizing the above discussion, we obtain the following lemma.

LEMMA 3.5.03:

Each number $L_{41}=1+2.4^i+3.4^j$, with $i+2 \leq j \leq m-1$, or with $j=m$ and $1 \leq i \leq m-1$, generates a different cyclotomic coset. All numbers of the form ν_{41} are contained in these cosets.

Concerning the numbers $\nu_{42}=1+3.4^i+2.4^j$, we have the following

LEMMA 3.5.04:

Each number $L_{42}=1+3.4^i+2.4^j$, with $i+1 \leq j \leq m-1$, or with $j=m$ and $1 \leq i \leq [m/2]$, generates a different cyclotomic coset. All numbers of the form ν_{42} are contained in these cosets.

Proof:

For the numbers $\nu_{42}=1+3.4^{i}+2.4^{j}$, we have

$$2\nu_{42,1} = 2 +2.4^{i} +1.4^{i+1} +1.4^{j+1}, \text{ if } j\neq m;$$
$$2\nu_{42,2} = 3 + 2.4^{i} + 1.4^{i+1}, \text{ if } j=m. \qquad (3.28a)$$

This means that, if $j \neq m$, then the cyclotomic coset containing $\nu_{42,1}$ has no other number of the form $\nu_{42,1}$. This leads that each number $L_{42,1}= 1+3.4^{i}+2.4^{j}$, with $j < m$, produces a different coset. In such a coset there is only one number of the form ν_{42}.

On the other hand, if $j = m$, we have

$$\nu_{42,2} = 1 + 3.4^{i} + 2.4^{m}$$
$$\nu'_{42,2} = 1 + 3.4^{m-i} + 2.4^{m}, \qquad (3.28b)$$

where $\nu'_{42,2}$ is a cyclic shift of $2\nu_{42,2}$. Comparing $\nu_{42,2}$ and $\nu'_{42,2}$, we easily obtain that each number $L_{42,2}= 1+3.4^{i}+2.4^{m}$, with $1 \leq i \leq [m/2]$, produces a different coset. Each of these cosets contains two numbers of the form ν_{42} except that, when m is even, there is one coset containing only one number of the form ν_{42}.

The number of numbers ν_{42} contained in the above cyclotomic cosets is given by

$$\begin{cases} C(m-1,2) + 2.[(m-1)/2] = C(m,2), & \text{if } m \text{ is odd;} \\ C(m-1,2)+2.[(m-2)/2]+1 = C(m,2), & \text{if } m \text{ is even.} \end{cases}$$

Since there are altogether $C(m,2)$ numbers of the form ν_{42}, we conclude that all numbers of the form ν_{42} are contained in the cosets of $L_{42,1}$ and $L_{42,2}$.

As regards the numbers of the form $\nu_{43}=3+3.4^i$, we have already seen in (3.25), while discussing the numbers of the form ν_{41}, that $\nu_{43,1} = 3+3.4^m$ and, therefore, $\nu'_{43,1} = 3+3.4^1$ are contained in the coset generated by $1+2.4^{m-1}+3.4^m$, a number of the form L_{41} in Lemma 3.5.03. Thus we have to consider only

$$\begin{aligned} \nu_{43,2} &= 3 + 3.4^i, \quad 1 < i < m, \\ 2\nu_{43,2} &= 2 + 1.4^1 + 2.4^i + 1.4^{i+1}. \end{aligned} \tag{3.29}$$

This means that no cyclic shift of $2\nu_{43,2}$ has the form of ν_{43}. On the other hand, $\nu'_{43,2}= 3+3.4^{m+1-i}$, which is a cyclic shift of $\nu_{43,2}$, has the form of ν_{43}.

The preceding discussion leads to the conclusion that each number $L_{43} = 3+3.4^i$, with $1 < i \leq [(m+1)/2]$, generates a different coset. All of these cosets contain two numbers of the form ν_{43} except that, when m is odd, there is one coset containing only one number of the form ν_{43}.

The number of numbers $\nu_{43,2}$ contained in the above cosets is given by

$$\begin{cases} 2.[(m/2)-1] = m-2, & \text{if m is even;} \\ 2.[(m+1)/2-2]+1 = m-2, & \text{if m is odd.} \end{cases}$$

Since there are altogether m-2 numbers of the form $\nu_{43,2}$, we conclude the following

LEMMA 3.5.05:

Each number $L_{43} = 3+3.4^i$, with $1 < i \leq [(m+1)/2]$, generates a different cyclotomic coset. All numbers of the form ν_{43} with $1 < i \leq m-1$ are contained in these cosets.

Next we consider the numbers of the form $\nu_{44} = 1+1.4^i+2.4^j+2.4^u$. We have the following

LEMMA 3.5.06:

Each number $L_{44}=1+1.4^{i}+2.4^{j}+2.4^{u}$, with $i+1 \le j \le [m/2]$ and $j+1 \le u \le m-1$, produces a different cyclotomic coset. All numbers ν_{44} with $u \le m-1$ are contained in these cosets.

Proof:

For the numbers $\nu_{44}=1+1.4^{i}+2.4^{j}+2.4^{u}$, we have

$$2\nu_{44,1} = 2+2.4^{i}+1.4^{j+1}+1.4^{u+1}, \text{ if } u \neq m;$$
$$2\nu_{44,2} = 3+2.4^{i}+1.4^{j+1}, \text{ if } u=m.$$

With reference to (3.28), we note that the numbers of the form $\nu_{44,2}$ have already been accounted in the cosets generated by L_{42} of Lemma 3.5.04. Therefore we have to consider only the numbers

$$\nu_{44,1} = 1+1.4^{i}+2.4^{j}+2.4^{u}, \quad u < m;$$
$$2\nu_{44,1} = 2+2.4^{i}+1.4^{j+1}+1.4^{u+1}.$$

On the basis of these two numbers we see that the cyclotomic coset generated by $\nu_{44,1}$ has the numbers, of the form ν_{44}, given by

$$\nu_{44,1} = 1+1.4^{i}+(2.4^{j})+2.4^{u}, \quad u < m;$$
$$\nu_{44,1} = 1+1.4^{u-j}+(2.4^{m-j})+2.4^{m-j+i}.$$

Comparing the bracketted terms in $\nu_{44,1}$ and $\nu'_{44,1}$, and following the previous methods on counting the numbers covered in the cyclotomic cosets, we easily obtain the stated result.

Finally, we pass on to the last case; namely, the numbers of the form $\nu_{45} = 1+2.4^{i} +1.4^{j}+2.4^{u}$. We have

$$2\nu_{45,1} = 2 +1.4^{i+1} +2.4^{j} +1.4^{u+1}, \text{ if } i+1\neq j,\ u\neq m;$$
$$2\nu_{45,2} = 3 + 1.4^{i+1} + 2.4^{j}, \text{ if } i+1\neq j,\ u=m;$$
$$2\nu_{45,3} = 2 + 3.4^{j} + 1.4^{u+1}, \text{ if } i+1=j,\ u\neq m;$$
$$2\nu_{45,4} = 3 + 3.4^{j}, \text{ if } i+1=j,\ u=m.$$

This means that the numbers of the form ν_{45}, for the cases other than $i+1\neq j$ and $u\neq m$, are all covered in the cyclotomic cosets considered previously. Specifically, with respect to (3.25)-(3.27), the numbers of forms $\nu_{45,2}$ and $\nu_{45,3}$ are covered in the cosets generated by L_{41} of Lemma 3.5.03. With respect to (3.29), the numbers of the form $\nu_{45,4}$ are covered in the cosets generated by L_{43} of Lemma 3.5.05. Thus we have to consider further only the numbers of the form

$$\nu_{45,1} = 1 +2.4^{i} +1.4^{j} +2.4^{u}, \quad i+1\neq j,\ u\neq m;$$
$$2\nu_{45,1} = 2 +1.4^{i+1} +2.4^{j} +1.4^{u+1}. \qquad (3.31)$$

Now let us consider the numbers $L_{45}= 1+2.4^{i}+1.4^{j}+2.4^{u}$, with $(j+i) \le u \le (m-i-1)$ for $(2i+1) \le j \le [(m+1)/2]$, and $(j+i+1) \le u \le (m-i-1)$ for $[(m+1)/2]+1 \le j \le (m-2i-2)$. From these constraints, it immediately follows that $1 \le i \le [(m-2)/4]$ for the former and $1 \le i \le [(m-6)/4]$ for the latter. The cyclotomic coset of L_{45} has three other numbers of the form ν_{45}; namely,

$$\begin{aligned} \nu'_{45,1} &= 1 +2.4^{u-j} +1.4^{m-j+1} +2.4^{m-j+i+1}, \\ \nu''_{45,1} &= 1 +2.4^{j-i-1} +1.4^{u-i} +2.4^{m-i}, \\ \nu'''_{45,1} &= 1 +2.4^{m-u} +1.4^{m-u+i+1} +2.4^{m-u+j}, \end{aligned} \tag{3.32}$$

where $\nu'_{45,1}$ is a cyclic shift of $\nu_{45,1}$, and $\nu''_{45,1}$ and $\nu'''_{45,1}$ are cyclic shifts of $2\nu_{45,1}$. From these numbers it can be shown that $\nu'_{45,1}$, $\nu''_{45,1}$ and $\nu'''_{45,1}$ are not members of $\{L_{45}\}$ except when $\nu'_{45,1}$ is with m+1 even and u-i=j=(m+1)/2. The cyclotomic cosets generated by L_{45} thus each contains two numbers of the form ν_{45} if j=(m+1)/2=u-i, and four numbers of the form ν_{45} otherwise.

To calculate the number of numbers of the form ν_{45} contained in the cosets generated by L_{45}, let us set $m-2 = 4(M+1)+\lambda$, where $\lambda \in \{0,1,2,3\}$, and M=0,1,2, ...

Then the number N_c of cyclotomic cosets produced by L_{45} is

$$N_c = \sum_{i=1}^{M+1} \sum_{j=2i+1}^{\lfloor (m+1)/2 \rfloor} (m-2i-j) + \sum_{i=1}^{M} \sum_{j=\lfloor (m+3)/2 \rfloor}^{m-2i-2} (m-2i-j-1)$$

Following certain manipulations, the above expression can be reduced to

$$N_c = \begin{cases} (M+1)(2M+1)(4M+3)/3, & \text{for } \lambda=0; \\ (M+1)[8(M+1)^2+1]/3, & \text{for } \lambda=1; \\ (M+1)(2M+3)(4M+5)/3, & \text{for } \lambda=2; \\ (M+1)[4(2M+3)(M+2)+3]/3, & \text{for } \lambda=3. \end{cases}$$

For m+1 odd, $\lambda=0$ or 2, N_c reduces to $C(m-2,3)/4$. Since each coset has four numbers of the form $v_{45,1}$, the number of numbers of the form $v_{45,1}$ contained in the cyclotomic cosets of L_{45} is $C(m-2,3)$. This is the total number of numbers of the form $v_{45,1}$. For m+1 even, $\lambda=1$ or 3, there are M+1 cosets each with two numbers of $v_{45,1}$ and $N_c-(M+1)$ cosets each with four numbers of $v_{45,1}$. Therefore, in the case of m+1 even, the number of numbers of the form $v_{45,1}$ contained in the cyclotomic cosets of L_{45} is

$$
\begin{aligned}
& 2(M+1) + 4[N_c-(M+1)]. \\
&= 4N_c - 2(M+1) \\
&= \begin{cases} \frac{4}{3}(M+1)[8(M+1)^2+1]-2(M+1), & \text{if } \lambda=1; \\ \frac{4}{3}(M+1)[4(2M+3)(M+2)+3]-2(M+1), & \text{if } \lambda=3; \end{cases} \\
&= \begin{cases} C(m-2,3), & \text{if } \lambda=1; \\ C(m-2,3)-1, & \text{if } \lambda=3. \end{cases}
\end{aligned}
$$

This means that, for $\lambda=1$, all numbers of the form $\nu_{45,1}$ are contained in the cosets produced by $\{L_{45}\}$; for $\lambda=3$, there is one more coset not covered by $\{L_{45}\}$. That coset is the one generated by $L'_{45} = 1 + 2.4^{(m-1)/4} + 1.4^{(m+1)/2} + 2.4^{(3m+1)/4}$. This coset has only one number of the form $\nu_{45,1}$.

Summarizing the above discussion, we conclude the following lemma.

LEMMA 3.5.07:

Each number $L_{45} = 1+2.4^{i} +1.4^{j} +2.4^{u}$, with $1 \le i \le [(m-2)/4]$, $2i+1 \le j \le [(m+1)/2]$ and $j+i \le u \le m-i-1$ on the one hand, and with $1 \le i \le [(m-6)/4]$, $[(m+1)/2]+1 \le j \le m-2(i+1)$ and $j+i+1 \le u \le m-i-1$ on the other hand, generates a different cyclotomic coset. In addition,

when $m=1 \bmod 4$, the number $L'_{45} = 1 + 2.4^{(m-1)/4} + 1.4^{(m+1)/2} + 2.4^{(3m+1)/4}$ generates another cyclotomic coset. All numbers of the form ν_{45}, with $i+1 \neq j$ and $u \neq m$, are contained in these cosets.

On the basis of Lemmas 3.5.01 to 3.5.07, we conclude the Theorem 3.5 below.

THEOREM 3.5:

For the case of $s=2$ and $m-r=2$, the cyclotomic cosets produced by the following representatives are disjoint, and the union of these cosets is the set S_ν:

$L_{31} = 1+4^i+4^j$, with $2i \leq j \leq m-i$;

$L'_{31} = 1+4^{(m+1)/3}+4^{2(m+1)/3}$,

with $m=5+3M$, $M=0,1,2,\ldots$;

$L_{32} = 1+2.4^i$, with $i=0,1,2,\ldots,[m/2]$;

$L_{41} = 1+2.4^i+3.4^j$, with $i+2 \leq j \leq m-1$,

or with $j=m$ and $1 \leq i \leq m-1$;

$L_{42} = 1+3.4^i+2.4^j$, with $i+1 \leq j \leq m-1$,

or with $j=m$ and $1 \leq i \leq [m/2]$;

$L_{43} = 3+3.4^i$, with $1 < i \leq [(m+1)/2]$;

$L_{44} = 1+4^{i}+2.4^{j}+2.4^{u}$, with $i+1 \le j \le [m/2]$ and $j+1 \le u \le m-1$;

$L_{45} = 1+2.4^{i}+1.4^{j}+2.4^{u}$, with

$j+i \le u \le m-i-1$ for $2i+1 \le j \le [(m+1)/2]$,

$j+i+1 \le u \le m-i-1$ for $[(m+1)/2]+1 \le j \le m-2(i+1)$;

$L'_{45} = 1+2.4^{(m-1)/4}+1.4^{(m+1)/2}+2.4^{(3m+1)/4}$, with $m=5+4M$, $M=0,1,2,\ldots$.

Example 3.6: Consider the case of $m+1=5$. With $s=2$ and $m-r=2$, the coset representatives corresponding to L_{31} are

$$1+4^{1}+4^{2}=21 \text{ and } 1+4^{1}+4^{3}=69;$$

those corresponding to L_{32} are 3, 9 and 33, as discussed in Example 3.4; those corresponding to L_{41} are

$$1+2.4+3.4^{3}=201,\ 1+2.4+3.4^{4}=777,$$
$$1+2.4^{2}+3.4^{4}=801,\ 1+2.4^{3}+3.4^{4}=897;$$

those corresponding to L_{42} are

$$1+3.4+2.4^{2}=45,\ 1+3.4+2.4^{3}=141,$$
$$1+3.4^{2}+2.4^{3}=177,\ 1+3.4+2.4^{4}=525,$$
$$1+3.4^{2}+2.4^{4}=561;$$

those corresponding to L_{43} and L_{44} are respectively the numbers

$$3+3.4^2=51 \text{ and } 1+4+2.4^2+2.4^3=165;$$

and finally, with $m+1=5$, there are no coset representatives corresponding to L'_{31}, L_{45} and L'_{45}. It is easily verified that all of the above numbers are in different cosets and that the radix-4 weights of these numbers and twice these numbers are 3 or 6.

3.4 DETAILS ABOUT CERTAIN CODES

Throughout our discussions on finding coset representatives, a few examples have been given to illustrate the essential points. However, we still think it is a good idea to give some more details here for several moderate length of projective geometry codes. The codes chosen for our demonstrations are the (1,3)th-order binary PG code of length 73 and the (2,2)th- and (1,2)th-order binary PG codes with the same length of 85. We hope that the illustrations here will make clear the essential ideas of this thesis.

3.4.1 (1,3)TH-ORDER PG CODE OF LENGTH 73

Consider the (1,3)th-order binary PG code of length 73, for which $m+1=3$, $r=1$, $s=3$, $q=2^s=8$, and $n=(q^{m+1}-1)/(q-1)=73$. This code has as its parity-check space all incidence vectors of the 1-flats in the projective geometry $PG(2,2^3)$, and is therefore 1-step majority-logic decodable.

Let α be a primitive element of $GF(8^3)$, and let ν be a nonnegative integer less than $q^{m+1}-1=511$. It follows that the generator polynomial of the (1,3)th-order binary PG code of length 73 has α^ν as a root if and only if ν is divisible by $q-1=7$ and

$$0 \le \max_{0\le \ell<3} W_8(2^\ell \nu \bmod 511) \le 7. \qquad (3.33)$$

With respect to the above condition, and following the discussion of Theorem 3.1, we deduce the nonnegative integers ν to be the numbers

$$\nu = 2^{i_0} + 2^{i_1} + 2^{i_2}, \qquad (3.34)$$

where $i_\ell \equiv \ell \bmod s=3$, $\ell=0,1,2$, and $0 \le i_\ell < s(m+1)=9$.

The integers less than $s(m+1)=9$ and congruent to 0, to 1, and to 2 modulo $s=3$ are:

$$\begin{aligned} 0,\ 3,\ 6 &\equiv 0 \mod 3 \\ 1,\ 4,\ 7 &\equiv 1 \mod 3 \\ 2,\ 5,\ 8 &\equiv 2 \mod 3. \end{aligned} \tag{3.35}$$

The numbers less than $q^{m+1}-1=511$ satisfying $W_8(\nu)=W_8(2\nu)=7$ are, according to (3.34),

$$\begin{aligned} &2^0+2^1+2^2,\ 2^0+2^4+2^2,\ 2^0+2^7+2^2, \\ &2^0+2^1+2^5,\ 2^0+2^4+2^5,\ 2^0+2^7+2^5, \\ &2^0+2^1+2^8,\ 2^0+2^4+2^8,\ 2^0+2^7+2^8; \\ \\ &2^3+2^1+2^2,\ 2^3+2^4+2^2,\ 2^3+2^7+2^2, \\ &2^3+2^1+2^5,\ 2^3+2^4+2^5,\ 2^3+2^7+2^5, \\ &2^3+2^1+2^8,\ 2^3+2^4+2^8,\ 2^3+2^7+2^8; \\ \\ &2^6+2^1+2^2,\ 2^6+2^4+2^2,\ 2^6+2^7+2^2, \\ &2^6+2^1+2^5,\ 2^6+2^4+2^5,\ 2^6+2^7+2^5, \\ &2^6+2^1+2^8,\ 2^6+2^4+2^8,\ 2^6+2^7+2^8. \end{aligned} \tag{3.36}$$

Since we are interested in odd numbers for coset representatives, we will consider only the first three rows of the above numbers. Representing the number $2^{i_0}+2^{i_1}+2^{i_2}$ with the set $\langle i_0,i_1,i_2\rangle$, we obtain the exponents of the odd numbers grouped by cosets:

$$\{<0,1,2>,\ <0,1,8>,\ <0,7,8>\},$$
$$\{<0,1,5>,\ <0,4,8>,\ <0,4,5>\},$$
$$\{<0,4,2>,\ <0,7,5>,\ <0,7,2>\}. \qquad (3.37)$$

Therefore, the nonzero coset representatives are

$$2^0 + 2^1 + 2^2 = 7,$$
$$2^0 + 2^1 + 2^5 = 35,$$
$$2^0 + 2^4 + 2^2 = 21. \qquad (3.38)$$

Alternatively, with reference to Theorem 3.4, we can get a sharp result without resorting complicated manipulations. Especially, according to Theorem 3.4, the coset representatives corresponding to L_{21} are the integers $1+6.8^1=49$ and $1+6.8^2=385$; the coset representative corresponding to L_{25} is $5+2.8^1=21$; and there are no coset representatives corresponding to L_{22}, L'_{22}, L_{26} and L'_{26}. Thus, the integers 21, 49 and 385 are nonzero coset representatives.

In the above two approaches, it should be noted that the integers $2^0+2^1+2^2=7$ and $1+6.8^2=385$ are actually in the same cyclotomic coset:

$$C_7 = \{\underline{7},14,28,56,112,224,448,\underline{385},259\},$$

and so are the integers $2^0+2^1+2^5=35$ and $1+6.8^1=49$:

$$C_{35} = \{\underline{35},70,140,280,\underline{49},98,196,392,273\}.$$

The integers 7, 21 and 35 are therefore the nonzero coset representatives useful for finding the generator polynomial of the discussed PG code.

Now, with reference to the Appendix C of Peterson and Weldon, we have, under degree s(m+1)=9, the minimal polynomial entries:

DEGREE 9

		1	1021E	3	1131E	5	1461G	7	1231A
9	1423G	11	1055E	13	1167F	15	1541E	17	1333F
19	1605G	21	1027A	23	1751E	25	1743H	27	1617H
29	1553H	35	1401C	37	1157F	39	1715E	41	1563H
43	1713H	45	1175E	51	1725G	53	1225E	55	1275E
73	0013	75	1773G	77	1511C	83	1425G	85	1267E.

Here, following each coset leader, the minimal function is given in an octal representation. Each digit of the octal notation can be transformed into three equivalent binary digits which are the coefficients of the polynomial, with the high-order coefficients at the left.

The minimal function corresponding to coset C_7 is the octal $(1231)_8$. The binary equivalent of $(1231)_8$ is 001 010 011 001, and hence the polynomial representation is

$$M_7(X) = 1 + X^3 + X^4 + X^7 + X^9.$$

The minimal function corresponding to C_{21} is $(1027)_8$, which has as binary equivalent 001 000 010 111, and hence the polynomial representation

$$M_{21}(X) = 1 + X + X^2 + X^4 + X^9.$$

The minimal function corresponding to C_{35} is $(1401)_8$, which has as binary equivalent 001 100 000 001, and hence the polynomial representation

$$M_{35}(X) = 1 + X^8 + X^9.$$

From the above minimal polynomial factors, the generator polynomial is calculated to be

$$\begin{aligned} g(X) &= (1+X).M_7(X).M_{21}(X).M_{35}(X) \\ &= (1+X).(1+X^3+X^4+X^7+X^9). \\ &\quad (1+X+X^2+X^4+X^9).(1+X^8+X^9). \end{aligned} \tag{3.39}$$

Hence the (1,3)th-order PG code of length 73 is a (73,45) cyclic code. Figure 3.2 gives a 28-stage shift-register encoding circuit for this code.

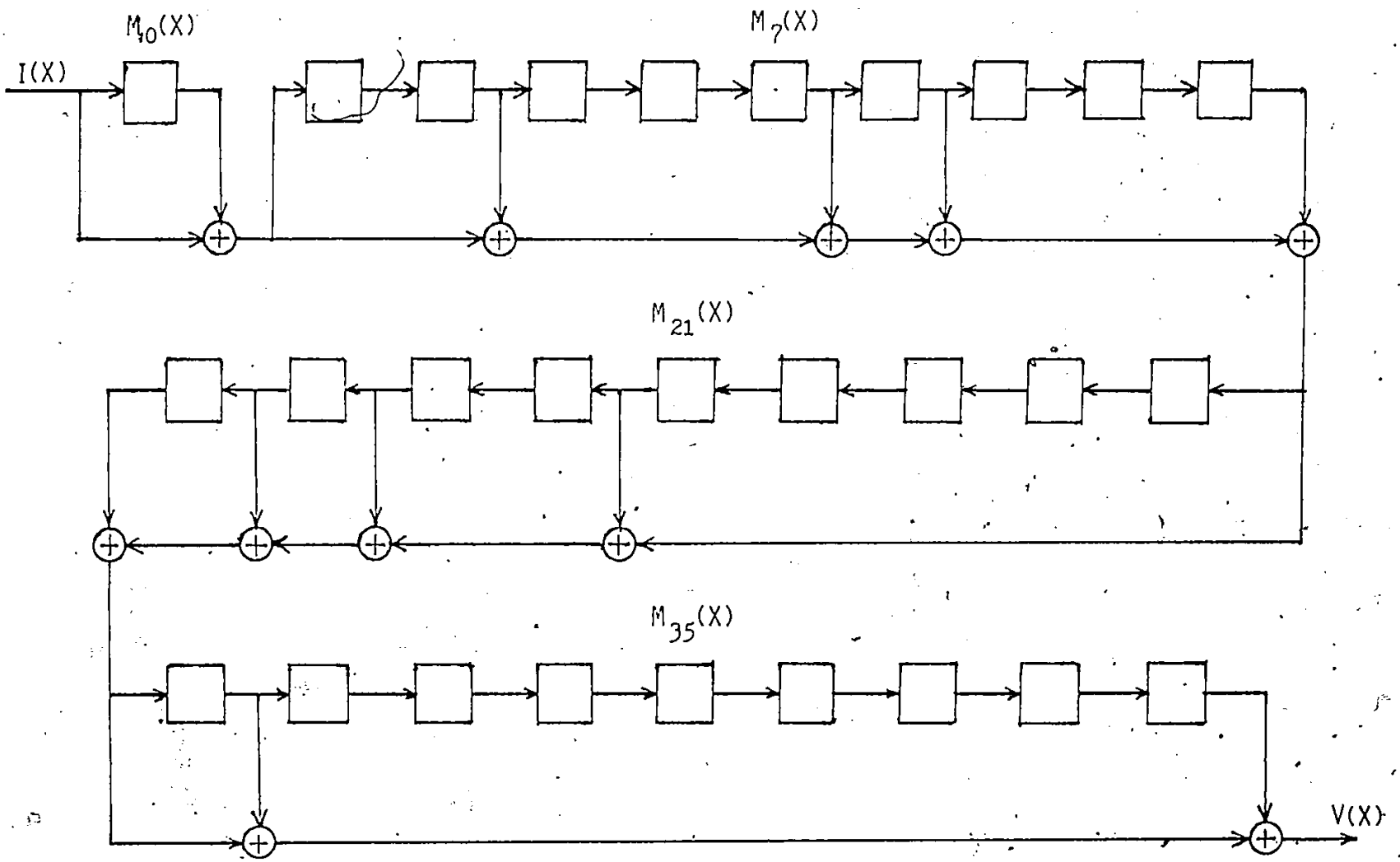


Figure 3.2 Shift register encoding circuit for (73,45) binary PG code.

3.4.2 (r,2)TH-ORDER PG CODE OF LENGTH 85

Let $m+1=4$, $r=2$, $s=2$ and $q=2^s=4$. Consider the (2,2)th-order binary PG code of length $n=(q^{m+1}-1)/(q-1)=85$. This code has as its parity-check space all the incidence vectors of the 1- and 2-flats in projective geometry $PG(3,2^2)$, and hence is 2-step majority-logic decodable.

Let α be the primitive element of $GF(2^8)$. Let ν be a nonnegative integer less than $2^{4\cdot 2}-1=255$. It follows that the generator polynomial of the (2,2)th-order binary PG code of length 85 has α^ν as a root if and only if ν is divisible by $2^2-1=3$ and

$$0 \le \max_{0\le \ell<2} W_4(2^\ell \nu \bmod 255) \le 3. \qquad (3.40)$$

The nonzero integers that satisfy the conditions above are, according to Theorem 3.1, the numbers

$$\nu = 2^{i_0} + 2^{i_1}, \qquad (3.41)$$

where $i_\ell \equiv \ell \bmod s=2$, $\ell=0,1$, and $0 \le i_\ell < s(m+1)=8$.

The integers less than $s(m+1)=8$ and congruent to 0 and to 1 modulo $s=2$ are:

$$\begin{aligned} &0,\ 2,\ 4,\ 6 \equiv 0 \mod 2 \\ &1,\ 3,\ 5,\ 7 \equiv 1 \mod 2. \end{aligned} \tag{3.42}$$

The odd numbers less than $q^{m+1}-1=255$ satisfying $W_4(\nu)=W_4(2\nu)=3$ are, according to (3.40),

$$\begin{aligned} &2^0+2^1=3, \quad 2^0+2^3=9, \\ &2^0+2^5=33, \quad 2^0+2^7=129. \end{aligned} \tag{3.43}$$

Representing the number $2^{i_0}+2^{i_1}$ with the set $<i_0,i_1>$, we obtain the exponents of the above odd numbers grouped by cosets to be

$$\begin{aligned} C_3 &= \{<0,1>,\ <0,7>\}; \\ C_9 &= \{<0,3>,\ <0,5>\}. \end{aligned} \tag{3.44}$$

Therefore, the coset representatives are $2^0+2^1=3$ and $2^0+2^3=9$.

With reference to the table of irreducible polynomials listed in the Appendix C of Peterson and Weldon, under the entry of degree $s(m+1)=8$, we can derive different minimal functions. The minimal function corresponding to coset C_3 is the octal $(567)_8$.

The binary equivalent of $(567)_8$ is 101 110 111, and hence the polynomial representation is

$$M_3(X) = 1 + X + X^2 + X^4 + X^5 + X^6 + X^8.$$

Similarly, the minimal function corresponding to coset C_9 is $(675)_8$, which has binary equivalent 110 111 101, and hence the polynomial representation

$$M_9(X) = 1 + X^2 + X^3 + X^4 + X^5 + X^7 + X^8.$$

Thus, the generator polynomial is

$$\begin{aligned} g_1(X) &= (1+X).M_3(X).M_9(X) \\ &= (1+X).(1+X+X^2+X^4+X^5+X^6+X^8). \\ &\quad (1+X^2+X^3+X^4+X^5+X^7+X^8) \end{aligned} \qquad (3.45)$$

implying that the code has $n-k=17$ parity-check bits. Hence, the (2,2)th-order PG code of length 85 is an (85,68) cyclic code. Figure 3.3 gives a 17-stage encoding circuit for this code. This circuit is in an alternative form of the multiply-by-g(X) circuit shown in Figure 3.1a.

At this point, we are also interested in finding the generator polynomial of the (1,2)th-order binary PG code of length 85. As we shall see, the generator

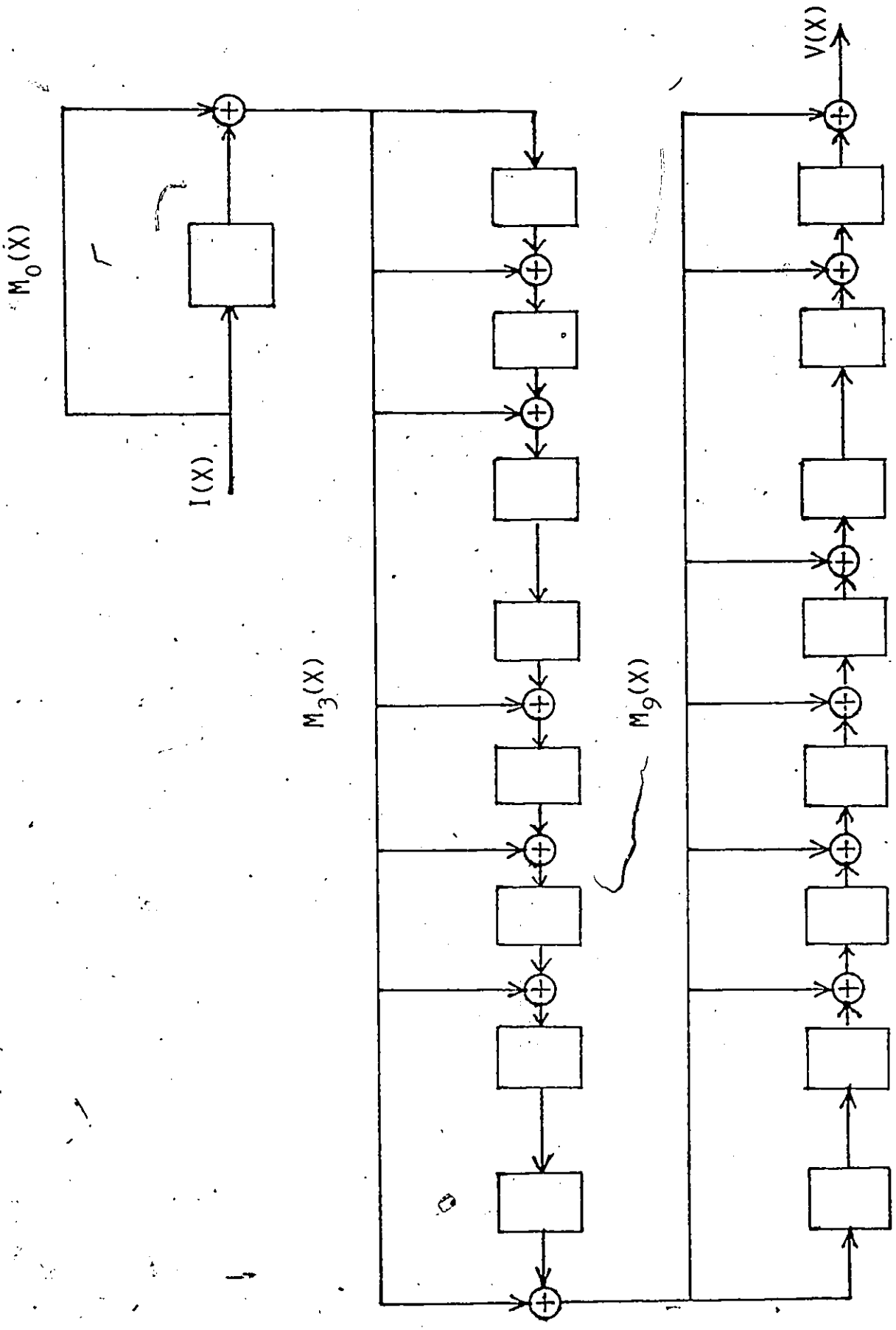


Figure 3.3 Shift register encoding circuit for (85,68) PG code.

polynomial of the (1,2)th-order PG code has the polynomial of (3.45) as a factor, and hence it forms a subcode of the (2,2)th-order binary PG code of the same length. Since the (1,2)th- PG code of length 85 has the parameters r=1, s=2, $q=2^s=4$, and $n=(q^{m+1}-1)/(q-1)=85$, we have m=3 and hence m-r=2. The discussions of Sections 3.1.2 and 3.3 can therefore be applied here.

According to Theorem 3.2, the odd numbers satisfying $\max_{0\le\ell<2} W_4(2^\ell \nu \bmod 255)= 3, 6$ are of the forms:

$$<0, i_1>,$$
$$<0, i_0', i_1, i_1'>,$$
$$<0, i_0', i_0''>,$$

where $i_\ell', i_\ell'' \equiv \ell \bmod 2$, $i_\ell' \neq i_\ell'' \neq 0$, and $i_\ell', i_\ell'' < 8$. The numbers corresponding to $<0,i_1>$ can be grouped into cosets as those given in (3.44); the numbers corresponding to $<0,i_0',i_1,i_1'>$ can be grouped into the cosets:

$$\begin{aligned}
C_{15} &= \{<0,2,1,3>, <0,2,1,7>, <0,6,1,7>, <0,6,5,7>\}\\
C_{39} &= \{<0,2,1,5>, <0,6,3,7>, <0,4,1,7>, <0,4,3,5>\}\\
C_{45} &= \{<0,2,3,5>, <0,2,5,7>, <0,6,1,3>, <0,6,3,5>\}\\
C_{27} &= \{<0,4,1,3>, <0,4,5,7>, <0,2,3,7>, <0,6,1,5>\}\\
C_{51} &= \{<0,4,1,5>, <0,4,3,7>\};
\end{aligned} \tag{3.46}$$

and the numbers corresponding to $<0,i_0',i_0''>$ can be grouped into the coset

$$C_{21} = \{<0,2,4>, <0,2,6>, <0,4,6>\}. \qquad (3.47)$$

Thus, the integers 3, 9, 15, 21, 27, 39, 45 and 51 are the nonzero coset representatives.

As an alternative, the coset representatives for the (1,2)th-order binary PG code of length 85 can be found through the approach depicted in Section 3.3. According to Theorem 3.5, the coset representative corresponding to L_{31} is the integer $1+4^1+4^2=21$; the coset representatives corresponding to L_{32} are the numbers $1+2.4^0=3$ and $1+2.4^1=9$; those corresponding to L_{41} are $1+2.4^1+3.4^3=201$ and $1+2.4^2+3.4^3=225$; those corresponding to L_{42} are $1+3.4^1+2.4^2=45$ and $1+3.4^1+2.4^3=141$; that corresponding to L_{43} is $3+3.4^2=51$; and there are no coset representatives corresponding to L_{31}', L_{44}, L_{45} and L_{45}'.

Among the above representatives, it can be verified that the numbers 201, 225 and 141 are respectively located in the cyclotomic cosets C_{39}, C_{15} and C_{27}. The generator polynomial of the (1,2)th- PG

code of length 85 is therefore the product of the minimal functions of $\alpha^{L_i} \in GF(2^8)$, with L_i= 0,3,9, 15,21,27, 39,45,51. That is,

$$g_2(X) = (1+X).M_3(X).M_9(X).M_{15}(X).M_{21}(X) .M_{27}(X).M_{39}(X).M_{45}(X).M_{51}(X), \quad (3.48)$$

which has $g_1(X)$ of (3.45) as a factor and is of degree 61, implying that there are 61 parity-check bits in the code word. Hence the (1,2)th-order binary PG code of length 85 is an (85,24) cyclic code.

Figure 3.4 gives a 24-stage systematic encoding circuit for the (1,2)th-order binary (85,24) PG code. Note that the tap-connections here are according to the coefficients of the parity-check polynomial

$$\begin{aligned} h_2(X) &= (1+X^{85})/g_2(X) \\ &= M_3^*(X).M_9^*(X).M_{21}^*(X) \\ &= 1 +X^5+X^7+X^9 +X^{10}+X^{12}+X^{13} \\ &\quad +X^{14}+X^{17}+X^{20} +X^{21}+X^{22}+X^{24}. \end{aligned} \quad (3.49)$$

This is a more economical choice than that using a 61-stage shift register circuit corresponding to the generator polynomial of (3.48).

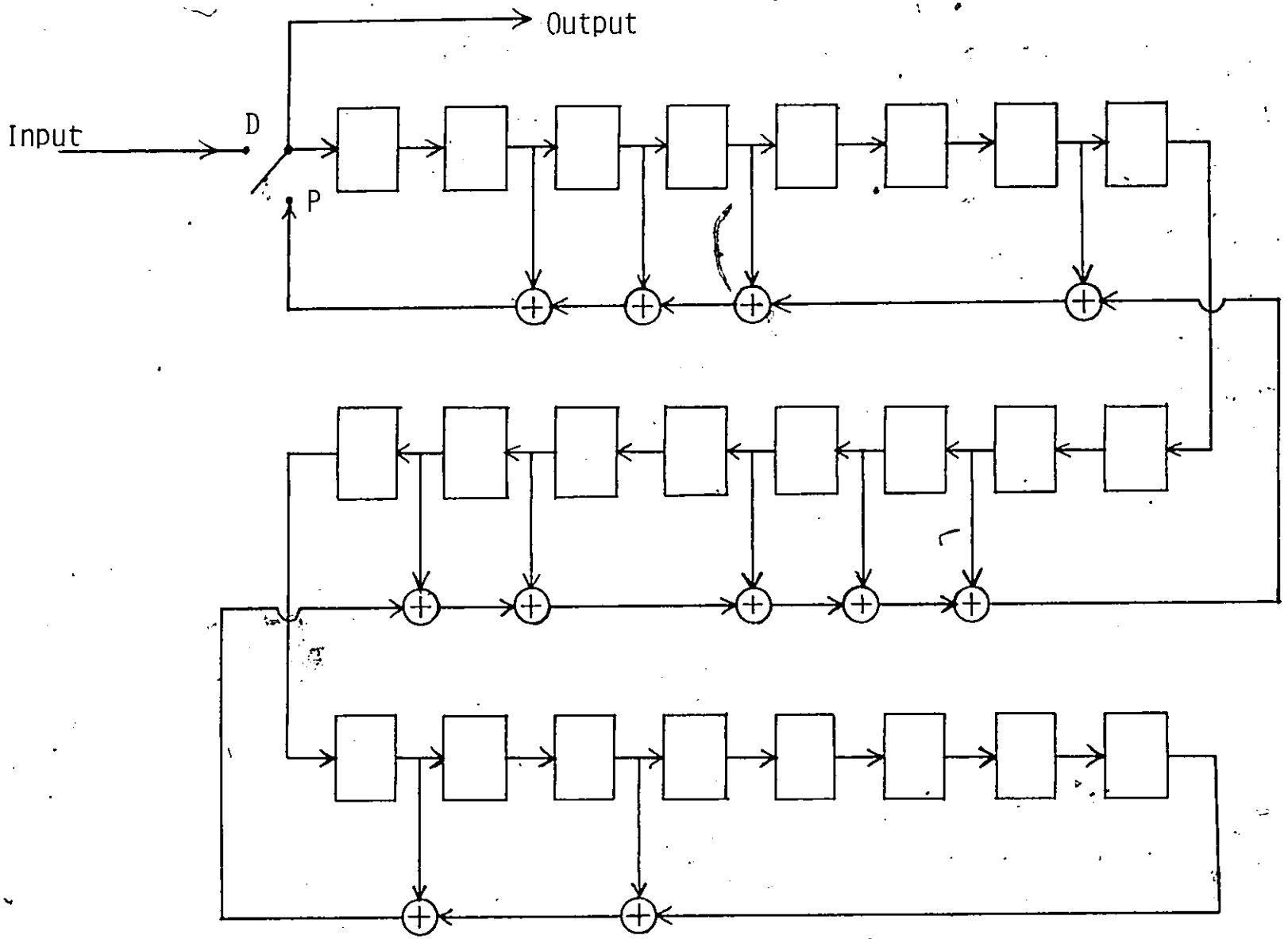


Figure 3.4 Systematic shift register encoder for (85,24) PG code.

Chapter IV

FINDING PARITY-CHECK SUMS FOR MAJORITY-LOGIC DECODING

Having investigated the coset representatives for finding the generator polynomials of binary projective geometry codes, we now turn to consider the problem of finding parity-check sums for decoding the code. Forming parity-check sums is central to the majority-logic decoding of projective geometry codes.

4.1 ORTHOGONAL CHECK SET AND MAJORITY-LOGIC DECODING

For a linear code, application of the parity-check rules on the received vector will produce a result which only depends on the error vector. The value resulting from the application of a parity-check rule on a received vector will be called a parity-check. By a symbol, we mean a coordinate of a vector, at a given position. When a parity check rule has a nonzero coefficient at a position corresponding to a given symbol, we shall say that the symbol is checked by that parity check rule.

An orthogonal check set of order J on a symbol is defined to be the set consisting of a collection of J parity check rules, each of which checks that particular symbol, and such that any other symbol is checked by at most one parity-check rule of the set. When the parity-check space contains orthogonal check sets of order J on each one of the n symbols, then the very simple majority-logic decoding method can be devised which permits the correction of any combination of [J/2] or fewer errors.

Considering the J parity-check rules that are orthogonal on the first symbol, we observe that, for any pattern of t errors, there are at most t among the J parity check rules which fail to be satisfied, whenever the first symbol is not corrupted by error, and at least J-(t-1) whenever one of the t errors affects the first symbol. Therefore, provided the number t of errors does not exceed [J/2], it can be decided by a majority vote on the J parity checks whether or not the first symbol was corrupted by error. Moreover, the actual value of the error is given by a clear majority of the J parity checks whenever the checking coefficient for the first symbol is 1 in each parity check rule. Now, applying

the same procedure on each of the n symbols will correct any combination of [J/2] or fewer errors. This procedure is known as one-step majority-logic decoding.

Rudolph [37] first used incidence vectors of points (0-flats) and lines (1-flats) in finite geometries to construct one-step majority-logic decodable codes. However, his decoding procedure was, in general, far from being able to correct the number of errors that the codes theoretically are able to correct. It turns out that another decoding procedure, known as multi-step majority-logic decoding, can be applied to these codes to correct a considerably greater number of errors. This decoding procedure, first introduced by Massey [27], will now briefly be described.

Suppose there exists, in the parity-check space of the code, a set of J parity-check rules, each of which checks a given subset of symbols with the same coefficients, and such that any other symbol is checked by at most one rule of the set, then this set of parity check rules is said to form an orthogonal check set of order J on the subset of symbols, and a new parity check set can be determined as follows: Let the coefficients

on the given subset of symbols be $c_{t_1}, c_{t_2}, \ldots, c_{t_\lambda}$. Then, with the same reasoning as above, provided the total number of errors does not exceed [J/2], the value of

$$c_{t_1}e_{t_1} + c_{t_2}e_{t_2} + \ldots + c_{t_\lambda}e_{t_\lambda}, \qquad (4.01)$$

where $e_{t_1}, e_{t_2}, \ldots, e_{t_\lambda}$ are the errors on the subset, can be determined by a clear majority of the J parity checks. Hence, the new parity check (4.01) can be determined. Now, suppose J new parity checks can be determined to form an orthogonal check set on a smaller subset. Then, in a second step, the same procedure will produce a new parity check with a smaller number of symbols involved. If this procedure can be applied several times until an orthogonal check set of order J has been constructed on a single symbol, then, at the last step, the actual value of the error on that symbol will be determined. If this can be done for each of the n symbols, in at most r steps, and if, at each step, the orthogonal check sets are of an order at least J, then the code is r-step majority-logic decodable for a number of errors not exceeding [J/2].

In terms of the above description, a general decoder for an r-step majority-logic decodable code is shown in Figure 4.1. The error correction procedure is as follows [23], [24]:

Step 1. The received vector $R(X)=r_0+r_1X+\ldots+r_{n-1}X^{n-1}$ is read into the buffer register.

Step 2. Parity-check sums orthogonal on certain properly selected sets of error digits are formed by summing appropriate sets of received digits. These check sums are then fed into the first-level majority-logic gates. The outputs of the first-level majority-logic gates are used to form inputs to the second-level majority gates. The outputs of the second-level majority-logic gates are then used to form inputs to the third-level majority gates. This continues until the last level is reached. The J inputs to the last-level gate are check sums orthogonal on the error digit e_0.

Step 3. The first received digit r_0 is read out of the buffer and is corrected by the output of the last-level majority gate.

Step 4. At the end of Step 3, the buffer register has been shifted one place to the right. Now the second received digit r_1 is in the first stage of the buffer register and will be corrected in exactly the same manner as the first received digit r_0 was. The decoder repeats Step 2 and Step 3.

Step 5. The received vector is decoded digit by digit in the manner described above for a total of n shifts.

With the above discussion, it is apparent that the formation of parity check sums plays a key role in the design of majority-logic decoder. For an (r,s)th-order projective geometry code, the parity-check sums are the incidence vectors of the r-flats in the projective geometry $PG(m,2^s)$. Hence, in the remainder of this chapter, we will devote our efforts on the construction of binary projective geometries.

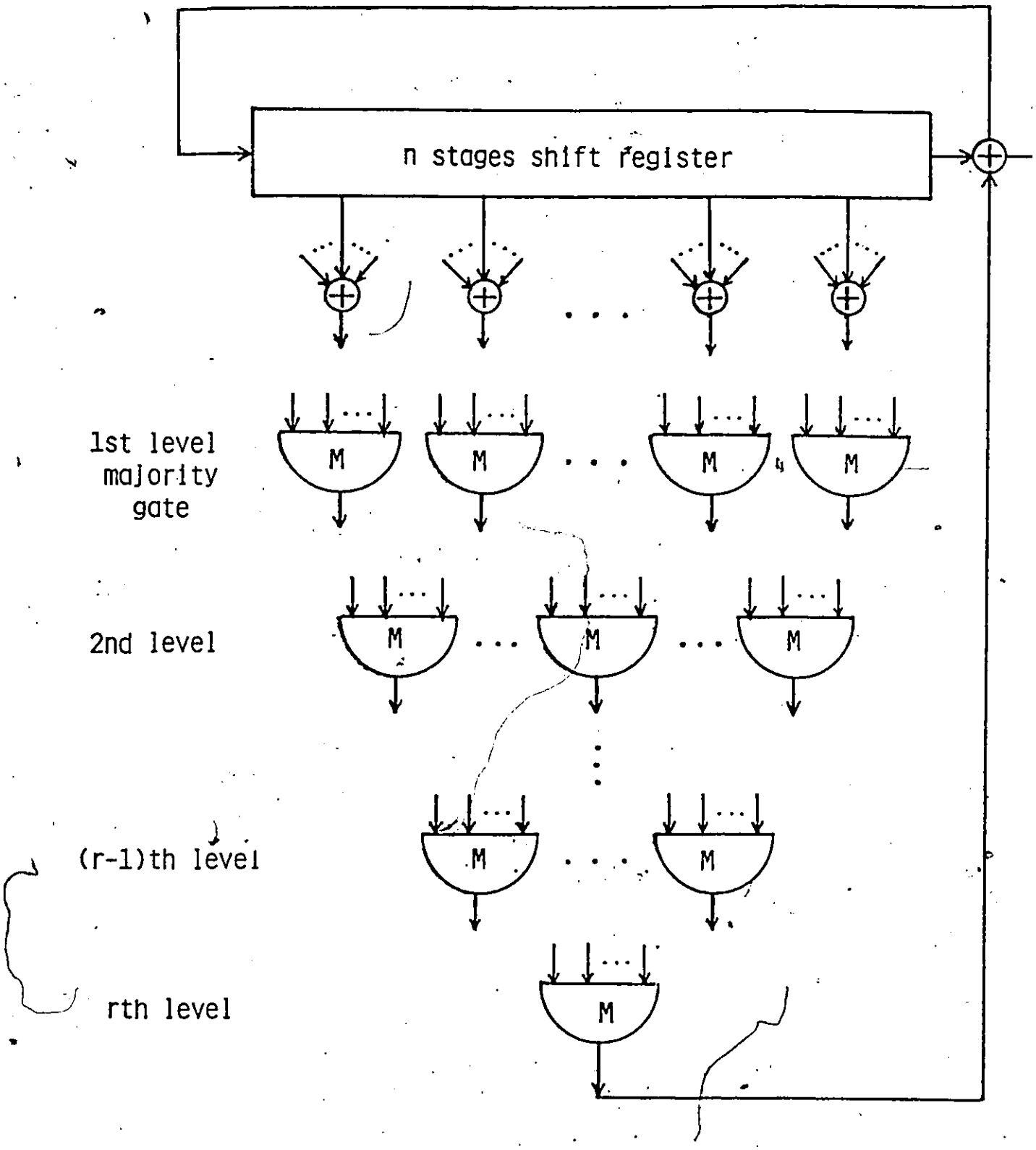


Figure 4.1 General r-step majority-logic decoder.

4.2 THE RECURSIVE FORMATION OF ORTHOGONAL CHECK SETS

The orthogonal parity-check sets of binary PG codes are associated with the structure of projective geometries. They can be obtained from the construction of orthogonal r-flats intersecting on an (r-1)-flat in PG(m,q), with r=1, 2, ..., m-1. This is done by first establishing a table list expressing all 0-flats of PG(m,q) in terms of combinations of the basis elements of $GF(q^{m+1})$. And then, according to the configuration of PG(m,q), by recursively connecting an (r-1)-flat and a point outside of it to form an r-flat. The details are as described below.

4.2.1 ESTABLISHING 0-FLATS OF PROJECTIVE GEOMETRY

Previously in Section 2.2, we mentioned the factorization of binary primitive polynomials into q-ary primitive polynomials. Especially, we gave a table list (Table 2.1) on polynomial factors over different Galois fields. With respect to this table, we note that the factors are monic polynomials of the form

$$P_i(X) = X^{m+1} + p_{im}X^m + \ldots + p_{i1}X + p_{i0}, \qquad (4.02)$$

where the coefficients p_{ij}'s are taken from GF(q). Corresponding to either of these factors, if α is a primitive element of $GF(q^{m+1})$, then the expression

$$\alpha^{m+1} = p_{im}\alpha^{m} + p_{i,m-1}\alpha^{m-1} + \ldots + p_{i1}\alpha + p_{i0} \qquad (4.03)$$

will provide a logic for connecting feedback taps of an (m+1)-stage shift register to generate the field elements of $GF(q^{m+1})$.

Precisely, we connect the linear feedback shift register in accordance with the logic (4.03), and load the memory cells with the initial value 0 0 0 ... 0 1. Then to the regular beat of a clock or other timing device, the contents of each memory cell will shift to the next cell. The instantaneous contents of the shift register will form a field element of $GF(q^{m+1})$, which is an (m+1)-tuple over GF(q). Since the points of the projective geometry PG(m,q) are defined to correspond to the nonzero elements of $GF(q^{m+1})$, with α^i, $\beta\alpha^i$, $\beta^2\alpha^i$, ..., and $\beta^{q-2}\alpha^i$ considered to be the same point, β being the primitive element of GF(q), $n=(q^{m+1}-1)/(q-1)$ clock times will be enough to generate the points of the projective geometry PG(m,q).

From the definition of an r-flat in a projective geometry, we know that the incidence vectors of all of the r-flats which contain a given (r-1)-flat define a set of parity checks that are orthogonal on the sum of the error digits associated with the points of that (r-1)-flat. For, given an (r-1)-flat $F^{(r-1)}$, any point (α^j) not belonging to $F^{(r-1)}$ is contained in precisely one r-flat that contains $F^{(r-1)}$. Hence, by the majority decoding algorithm described before, a new parity check can be obtained that corresponds to the incidence vector of the (r-1)-flat $F^{(r-1)}$. This can be similarly done for all of the (r-1)-flats that contain a given (r-2)-flat, which in turn define a set of parity checks orthogonal on that (r-2)-flat. Hence, by induction, we obtain after r steps a set of parity checks orthogonal on a 0-flat, that is, on a single error symbol.

Based on the above observation, we propose the following construction of r-flats in PG(m,q). In this approach, a scratch pad of memory is needed for remembering the location of any single point within the constructed r-flat. For convenience, without loss of generality, we will take the 0-flat (α^0) as our object on forming r-flats orthogonal to it.

4.2.2 COMPUTER ALGORITHM ON FORMING ORTHOGONAL FLATS

To form orthogonal flats, we first select two distinct points other than (α^0), say the points (α^{i_1}) and (α^{i_2}). These two points will form a line consisting of points of the form $(a_1\alpha^{i_1}+a_2\alpha^{i_2})$, where a_1 and a_2 are from GF(q)= $\{0,1,\beta,\beta^2, \ldots, \beta^{q-2}\}$, and a_1 and a_2 are not both equal to zero. In other words, the line passing through (α^{i_1}) and (α^{i_2}) contains the points

$$(\alpha^{i_1}),(\alpha^{i_2}),(\alpha^{i_1+\alpha^{i_2}}),(\alpha^{i_1+\beta\alpha^{i_2}}),$$
$$(\alpha^{i_1+\beta^2\alpha^{i_2}}), \ldots, (\alpha^{i_1+\beta^{q-2}\alpha^{i_2}}). \qquad (4.04)$$

By arranging the above points in different 1-flats (as shown by the marks X in Figure 4.2a), we can connect each of them to the point (α^0) (the mark Δ in Figure 4.2a) and form q+1 1-flats orthogonal on (α^0):

$$F_1' = \{(\alpha^0), (\alpha^{i_1}), (\alpha^0+\alpha^{i_1}), (\alpha^0+\beta\alpha^{i_1}),$$
$$(\alpha^0+\beta^2\alpha^{i_1}), \ldots, (\alpha^0+\beta^{q-2}\alpha^{i_1})\};$$

$$F_2' = \{(\alpha^0), (\alpha^{i_2}), (\alpha^0+\alpha^{i_2}), (\alpha^0+\beta\alpha^{i_2}),$$
$$(\alpha^0+\beta^2\alpha^{i_2}), \ldots, (\alpha^0+\beta^{q-2}\alpha^{i_2})\};$$

$$F'_j = \{(\alpha^0),\ (\alpha^{i1}+\alpha^{i2}),\ (\alpha^0+(\alpha^{i1}+\alpha^{i2})),$$
$$(\alpha^0+\beta(\alpha^{i1}+\alpha^{i2})),\ \ldots,$$
$$(\alpha^0+\beta^{q-2}(\alpha^{i1}+\alpha^{i2}))\};$$

$$F'_{q+1} = \{(\alpha^0),\ (\alpha^{i1}+\beta^{q-2}\alpha^{i2}),$$
$$(\alpha^0+(\alpha^{i1}+\beta^{q-2}\alpha^{i2})),$$
$$(\alpha^0+\beta(\alpha^{i1}+\beta^{q-2}\alpha^{i2})),\ \ldots,$$
$$(\alpha^0+\beta^{q-2}(\alpha^{i1}+\beta^{q-2}\alpha^{i2}))\}. \qquad (4.05)$$

For more than one step majority-logic decodable PG codes, once the q+1 1-flats orthogonal on a 0-flat are formed, our next task is to form q+1 2-flats orthogonal on each of these 1-flats. Apparently, the q+1 1-flats of (4.05) can be united into a 2-flat. This 2-flat contains either one of the ingredient 1-flats. In the following we will use the notation F''_{ij}, i,j=1, 2, ..., q+1, to denote the 2-flats orthogonal on the 1-flat F'_i.

On constructing 2-flats F''_{ij}'s orthogonal on the 1-flat F'_i, the first one can be simply the union of F'_1 to F'_{q+1}:

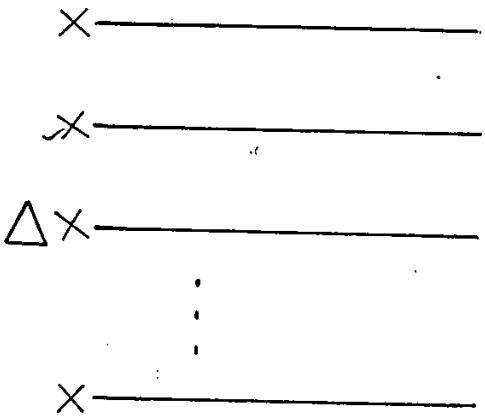
$$F''_{i1} = F'_1 \cup F'_2 \cup \ldots \cup F'_{q+1}. \qquad (4.06)$$

This 2-flat surely contains the ingredient 1-flat F_i'. Now suppose the point (α^{j_2}) is not in F_{i1}''. Then the line passing through (α^{j_2}) and $(\alpha^{j_1}) \in F_{i1}''-F_i'$ is the set made up of points of the form $(a_1\alpha^{j_1}+a_2\alpha^{j_2})$, with $a_1,a_2 \in GF(q)$ and are not both zero. That is, we have the following points:

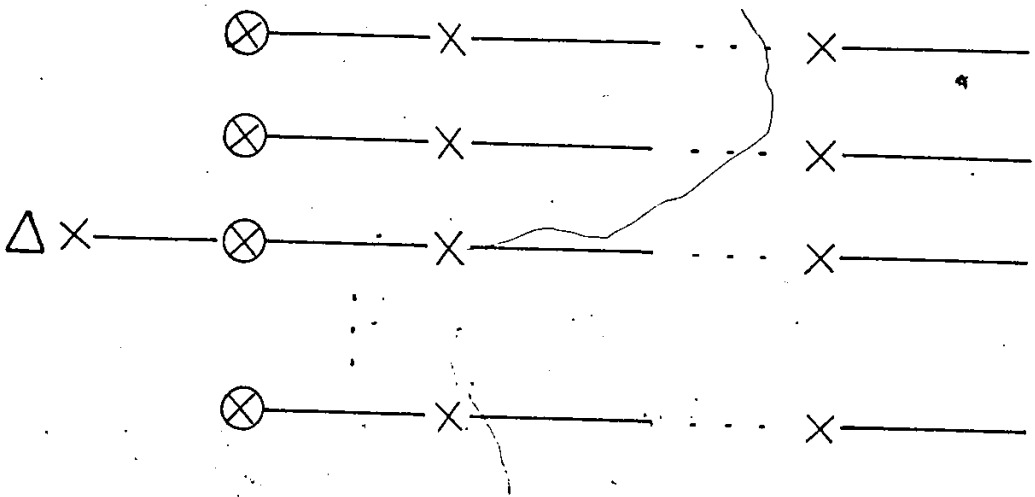
$$(\alpha^{j_1}),(\alpha^{j_2}),(\alpha^{j_1}+\alpha^{j_2}),(\alpha^{j_1}+\beta\alpha^{j_2}),$$
$$(\alpha^{j_1}+\beta^2\alpha^{j_2}), \ldots, (\alpha^{j_1}+\beta^{q-2}\alpha^{j_2}). \qquad (4.07)$$

Each of these points is located in different 2-flats (the ⊗ marks in Figure 4.2b). Hence we can associate each of them with the 1-flat F_i' and form q+1 orthogonal 2-flats.

To associate each of the points in (4.07) with the 1-flat F_i', we refer back to the procedures on forming 1-flats orthogonal to the 0-flat (α^0). That is, with $(\alpha^{k_1}) \in F_i'-(\alpha^0)$ and with (α^{k_2}) being either of the points in (4.07), we locate q+1 points (the X marks in each row of Figure 4.2b) on the line connecting (α^{k_1}) and (α^{k_2}); then we associate (α^0) with each of these points to form q+1 1-flats. The union of these 1-flats is a 2-flat containing the 1-flat F_i'.



(a) Scratch pad for constructing 1-flats orthogonal to the 0-flat Δ.



(b) Scratch pad for constructing 2-flats orthogonal to the 1-flat denoted by ΔX———.

Figure 4.2 Schematic diagram on forming orthogonal flats.

In the above manner, for each 1-flat F'_i, i=1, 2, ..., q+1, we can form q+1 2-flats F''_{i1} --- $F''_{i,q+1}$ orthogonal on it. Figure 4.2b depicts a schematic diagram on constructing J=q+1 2-flats orthogonal to a specified 1-flat.

For more than two steps majority-logic decodable PG codes, we have to go further to construct 3-flats orthogonal on each of the above 2-flats. Let F'''_{iju}, i,j,u = 1, 2, ..., q+1, denote a 3-flat containing the 2-flat F''_{ij}, which in turn contains the 1-flat F'_i, and which in turn contains the 0-flat (α^0).

On constructing the 3-flats F'''_{iju}'s orthogonal to the 2-flat F''_{ij}, the first one can be simply the union of F''_{i1} to $F''_{i,q+1}$:

$$F'''_{ij1} = F''_{i1} \cup F''_{i2} \cup \ldots \cup F''_{i,q+1}. \qquad (4.08)$$

This 3-flat surely contains the ingredient 2-flat F''_{ij}. Now suppose the point (α^{u_2}) is not in F'''_{ij1}. Then on the line passing through (α^{u_2}) and $(\alpha^{u_1}) \in F'''_{ij1}-F''_{ij}$ we can locate q+1 points:

$$(\alpha^{u_1}),(\alpha^{u_2}),(\alpha^{u_1}+\alpha^{u_2}),(\alpha^{u_1}+\beta\alpha^{u_2}),$$
$$(\alpha^{u_1}+\beta^2\alpha^{u_2}),\ \ldots,\ (\alpha^{u_1}+\beta^{q-2}\alpha^{u_2}). \qquad (4.09)$$

Each of these points is in different 3-flats (the ⊗ marks in Figure 4.3), so they can be each associated with the 2-flat F''_{ij} to result in q+1 orthogonal 3-flats. This is done by recursively referring back to the procedures on constructing 2- and 1-flats.

To associate F''_{ij} with either one of the points in (4.09), say the point (α^{v_2}), we choose a point (α^{v_1}) in $F''_{ij}-F'_i$ and connect it to (α^{v_2}). This results in a line consisting of q+1 points (the ⊗ marks in each row of Figure 4.3); each can then associate with the 1-flat F'_i to result in q+1 2-flats orthogonal on F'_i. This is done by referring to the procedures on constructing 1-flats. That is, with $(\alpha^{w_1}) \in F'_i-(\alpha^0)$ and with (α^{w_2}) being a point on the line connecting (α^{v_1}) and (α^{v_2}), we connect them to form a line (the X marks in each sub-column of Figure 4.3). Then we associate each point of the line with the 0-flat (α^0) and form q+1 orthogonal 1-flats. The union of these 1-flats is a 2-flat. The union of the above q+1 2-flats is a 3-flat.

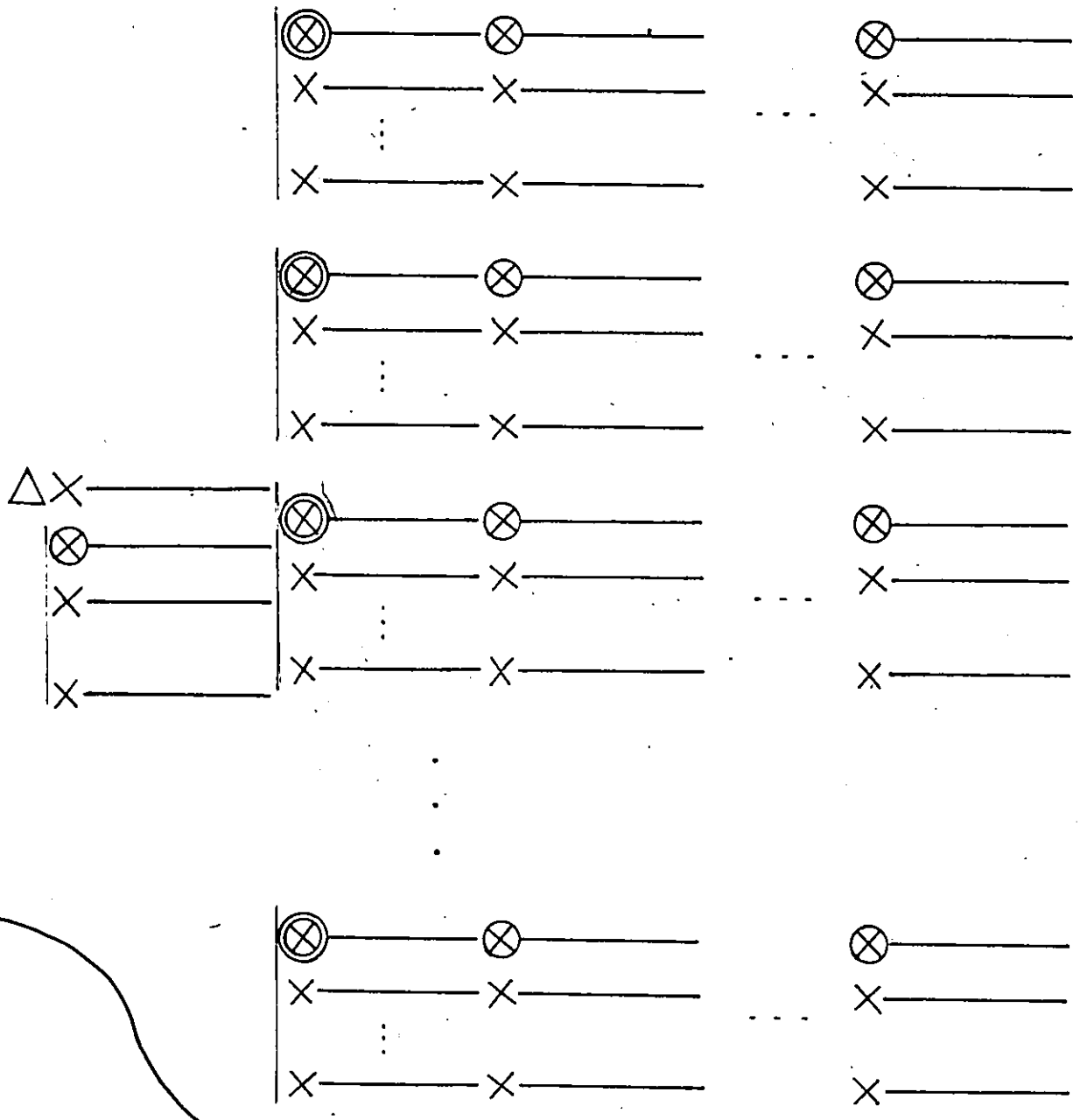


Figure 4.3 Scratch pad for constructing 3-flats orthogonal to a 2-flat depicted at the leftmost column.

In the above manner, for each 1-flat F_i^1, i=1, 2, ..., q+1, we can form q+1 2-flats F_{i1}^2 --- $F_{i,q+1}^2$ orthogonal on it. For each 2-flat F_{ij}^2 containing the 1-flat F_i^1, j=1,2, ..., q+1, we can form q+1 3-flats F_{ij1}^3 --- $F_{ij,q+1}^3$ orthogonal on it. Figure 4.3 depicts a schematic diagram on constructing J=q+1 3-flats orthogonal to a specified 2-flat.

For (m-1)-step majority-logic decodable PG codes, by recursively applying the above procedures, we can form J=q+1 1-flats orthogonal on a 0-flat, J=q+1 2-flats orthogonal on each of these q+1 1-flats, J=q+1 3-flats orthogonal on each of the above $(q+1)^2$ 2-flats, and so on. At the end of the recursions, there are J=q+1 (m-1)-flats intersecting on each of the $(q+1)^{m-2}$ (m-2)-flats.

As for the (m-2)-step majority-logic decodable binary PG code of length $n=(q^{m+1}-1)/(q-1)$, the orthogonal parity-check sets can be simply derived from those of the (m-1,s)th-order binary PG code of the same length. Recall that, in the (m-1)-step majority-logic decodable PG code, at the (m-μ)th decoding step, μ=1, 2, ..., m-2, there are J=q+1 (m-μ)-flats orthogonal on

each of the q+1 (m-μ-1)-flats which are orthogonal on an (m-μ-2)-flat. The first one of these J=q+1 (m-μ)-flats is the union of the q+1 (m-μ-1)-flats formed for the (m-μ-1)th decoding step of the (m-1,s)th-order PG code of length n.

Each of the above J=q+1 (m-μ)-flats can be decomposed into q+1 (m-μ-1)-flats. Totally, we will get $(q+1)^2$ (m-μ-1)-flats. However, the one the J=q+1 (m-μ)-flats are orthogonal with is being counted q+1 times. Hence we have actually $(q+1)^2-(q+1)+1 = (q^3-1)/(q-1)$ different (m-μ-1)-flats which are intersecting on an (m-μ-2)-flat. These (m-μ-1)-flats can be taken to form a set of $J'=(q^3-1)/(q-1)$ orthogonal check sums for the (m-2,s)th-order binary PG code of length n.

Further proceed the above procedures on the (m-2)-step majority-logic decodable PG code of length $n=(q^{m+1}-1)/(q-1)$, each of the $J'=(q^3-1)/(q-1)$ (m-μ-1)-flats can be decomposed into q+1 (m-μ-2)-flats. Totally, there are $(q^4-1)/(q-1)$ different (m-μ-2)-flats which are orthogonal on an (m-μ-3)-flat. These (m-μ-2)-flats can then be taken to form $J''=(q^4-1)/(q-1)$

orthogonal check sums for the (m-3,s)th- binary PG code of length n.

In general, by recursively applying the above decomposition procedures to an (r,s)th-order binary PG code of length $n=(q^{m+1}-1)/(q-1)$, r=m-1, m-2, ..., 3, 2, each of the $J=(q^{m-r+1}-1)/(q-1)$ $(r-\mu+1)$-flats orthogonal on an $(r-\mu)$-flat, $\mu=1, 2, \ldots, r-1$, can be decomposed into q+1 $(r-\mu)$-flats. Among these J(q+1) $(r-\mu)$-flats, the one the $J=(q^{m-r+1}-1)/(q-1)$. $(r-\mu+1)$-flats are orthogonal with is being counted J times. Hence we have actually $Jq+1= (q^{m-r+2}-1)/(q-1)$ $(r-\mu)$-flats which are intersecting on an $(r-\mu-1)$-flat. These $(r-\mu)$-flats can be taken to form $J'= Jq+1= (q^{m-r+2}-1)/(q-1)$ orthogonal check sums suitable for the (r-1,s)th-order binary PG code of length n.

4.3 SEQUENTIAL CODE REDUCTION (SCR) DECODING

For large values of J and r, the conventional rule of orthogonalization for decoding long PG codes requires a very large number of majority-logic gates, and thus the decoding complexity becomes prohibitive. However,

if we are willing to sacrifice the decoding speed, the decoding complexity may be reduced.

For an r-step majority-logic decodable PG code, only one J-input majority-logic gate is needed at each level of orthogonalization. The successive check sums formed at the output of the μth-level majority-logic gate, $1 \leq \mu \leq r-1$, are first stored in a buffer register and then combined to form J check sums that are orthogonal on a properly selected set of error digits. These J orthogonal check sums are inputs to the $(\mu+1)$th-level majority-logic gate. The successive check sums formed at the output of this $(\mu+1)$th-level gate are again stored and combined to form J check sums orthogonal on another properly selected set of error digits. These J new orthogonal check sums form the inputs to the $(\mu+2)$th-level majority-logic gate. This process continues until the last level of orthogonalization is reached; the output of the last-level majority-logic gate is the estimated value of a specific error digit.

The majority-logic decoding algorithm described above is called sequential-code-reduction (SCR)

majority-logic decoding. This decoding scheme was first proposed by Rudolph [38] and has brought a new research direction on the decoding complexity [39]. Using this decoding scheme, only r majority-logic gates are required for an r-step SCR majority-logic decodable code, one majority-logic gate at each level of orthogonalization. A buffer register of no more than n stages is needed between every two consecutive levels of orthogonalization.

4.3.1 GENERATING ORTHOGONAL FLAT

Consider the decoding of the lowest-order error digit e_0 with sequential code reduction technique. Let

$$\theta_0 = e_0 + e_{t_1} + e_{t_2} + \ldots + e_{t_\lambda} \qquad (4.10)$$

be the first check sum formed at the output of the $(\mu-1)$th-level majority-logic gate, where $0 < t_1 < t_2 < \ldots < t_\lambda$. Let

$$f(X) = 1 + X^{t_1} + X^{t_2} + \ldots + X^{t_\lambda} \qquad (4.11)$$

be the orthogonal polynomial associated to θ_0. At the successive clock times, the outputs of the $(\mu-1)$th-level majority-logic gate are the following check sums:

$$\theta_1 = e_1 + e_{t_1+1} + e_{t_2+1} + \cdots + e_{t_\lambda+1},$$
$$\theta_2 = e_2 + e_{t_1+2} + e_{t_2+2} + \cdots + e_{t_\lambda+2},$$
$$\vdots$$
$$\theta_\ell = e_\ell + e_{t_1+\ell} + e_{t_2+\ell} + \cdots + e_{t_\lambda+\ell},$$
$$\vdots$$
$$\theta_{n-1} = e_{n-1} + e_{t_1+n-1} + \cdots + e_{t_\lambda+n-1}. \qquad (4.12)$$

Note that the indices of the error digits are cyclically shifted to the right one place every clock time (if $t_j+\ell$ is $\geq n$, it is replaced by $t_j+\ell-n$). The polynomial associated with the check sum θ_ℓ is therefore

$$f^{(\ell)}(X) = X^\ell + X^{t_1+\ell} + X^{t_2+\ell} + \cdots + X^{t_\lambda+\ell}, \qquad (4.13)$$

which is obtained by cyclically shifting $f(X)$ to the right ℓ times. In fact, $f^{(\ell)}(X)$ is the remainder resulting from dividing $X^\ell f(X)$ by $1+X^n$. That is,

$$X^\ell f(X) = Q_\ell(X)(1+X^n) + f^{(\ell)}(X);$$

or equivalently,

$$f^{(\ell)}(X) = X^\ell f(X) + Q_\ell(X)(1+X^n)$$
$$= X^\ell f(X) \bmod (1+X^n). \qquad (4.14)$$

Let the first check sum formed at the output of the μth-level majority-logic gate be

$$S_1 = e_0 + e_{t_1'} + e_{t_2'} + \ldots + e_{t_\emptyset'}. \qquad (4.15)$$

Using SCR majority-logic decoding, the check sums orthogonal on S_1 are linear combinations of θ_0, θ_1, θ_2, ..., θ_{n-1}. Let S_i' be a check sum orthogonal on S_1. Then

$$S_i' = c_{i0}\theta_0 + c_{i1}\theta_1 + \ldots + c_{i,n-1}\theta_{n-1}. \qquad (4.16)$$

Let $f_i'(X)$ be the orthogonal polynomial associated with the error digits in S_i'. Then we have

$$f_i'(X) = c_{i0}f(X) + c_{i1}f^{(1)}(X) + \ldots + c_{i,n-1}f^{(n-1)}(X). \qquad (4.17)$$

Combining (4.14) and (4.17), we obtain

$$f_i'(X) = (c_{i0}+c_{i1}X+\ldots+c_{i,n-1}X^{n-1})f(X) = c_i(X)f(X) \bmod (1+X^n). \qquad (4.18)$$

From (4.18) we see that each orthogonal polynomial $f_i'(X)$ at the μth-level of orthogonalization is a multiple of $f(X)$ with operation taken modulo $1+X^n$.

The polynomial $f(X)$ is called the generating orthogonal flat. At each level of SCR majority-logic

decoding, if we know $f(X)$ and $f_i'(X)$ and if we are able to express $f_i'(X)$ in the form of (4.18), we can form check sums S_i' orthogonal on S_1 from (4.16). Therefore, a cyclic code is SCR majority-logic decodable if and only if there exists a generating orthogonal flat at each level of orthogonalization. In other words, a necessary and sufficient condition for the existence of an r-stage sequential-code-reduction algorithm is that at each stage of majority-logic decoding there exists a flat that divides each member of a set of J flats orthogonal on a flat of lower dimension at the next stage [38].

4.3.2 FINDING TAP COEFFICIENTS FOR SCR DECODING

Finding the polynomial $C_i(X)$ to satisfy (4.18) is equivalent to finding polynomials $C_i(X)$ and $D_i(X)$ such that

$$f_i'(X) = C_i(X)f(X) + D_i(X)(1+X^n). \qquad (4.19)$$

To do this, we have to resort to the famous Euclidean algorithm [1], [28].

EUCLIDEAN ALGORITHM:

For any two polynomials a(X) and b(X) with coefficients from some given field, there exist polynomials s(X) and t(X) such that

$$s(X)a(X) + t(X)b(X) = (a(X),b(X)),$$

where (a(X),b(X)) denotes the greatest common divisor (GCD) of a(X) and b(X).

The above algorithm involves four sequences of polynomials: $(s_i(X))$, $(t_i(X))$, $(r_i(X))$, and $(q_i(X))$. The initial conditions are

$$s_{-1}(X)=1,\ t_{-1}(X)=0,\ r_{-1}(X)=a(X),$$
$$s_0(X)=0,\ t_0(X)=1,\ r_0(X)=b(X). \qquad (4.20)$$

For $i \geq 1$, $q_i(X)$ and $r_i(X)$ are defined to be the quotient and remainder, respectively, when $r_{i-2}(X)$ is divided by $r_{i-1}(X)$:

$$r_{i-2}(X) = q_i(X)r_{i-1}(X) + r_i(X),$$
$$\deg r_i(X) < \deg r_{i-1}(X). \qquad (4.21)$$

The polynomials $s_i(X)$ and $t_i(X)$ are then defined by

$$s_i(X) = s_{i-2}(X) - q_i(X)s_{i-1}(X),$$
$$t_i(X) = t_{i-2}(X) - q_i(X)t_{i-1}(X). \qquad (4.22)$$

Since the degrees of the remainders $r_i(X)$ are strictly decreasing, there will be a last nonzero one; call it $r_z(X)$. It turns out that $r_z(X)$ is the greatest common factor of a(X) and b(X), and furthermore that $s_z(X)a(X) + t_z(X)b(X) = r_z(X)$.

Now, suppose g(X) and h(X) are the generator and the parity-check polynomials of binary PG subcode, with g*(X) and h*(X) being the respective reciprocals. For f(X) being the generating orthogonal flat in the dual of the subcode, we have

$$GCD(f(X), 1+X^n) = h^*(X), \qquad (4.24)$$

where $h^*(X) = (1+X^n)/g^*(X)$ is the generator polynomial of the dual subcode. The generating orthogonal flat f(X) is therefore a multiple of h*(X),

$$f(X) = h^*(X)I_f(X). \qquad (4.25)$$

Also, the flat $f_i'(X)$ is a codeword in the dual of the subcode, and hence is a multiple of h*(X),

$$f_i'(X) = h^*(X)I_i'(X). \qquad (4.26)$$

Equation (4.19) can therefore be written into the form

$$C_i(X)h^*(X)I_f(X) + D_i(X)h^*(X)g^*(X) = h^*(X)I_i'(X). \quad (4.27)$$

In the above expression, we note that $I_f(X)$ and $g^*(X)$ are relatively prime; otherwise the greatest common divisor of $f(X)$ and $1+X^n$ will not be $h^*(X)$, and $f(X)$ will not be the generating orthogonal flat of the dual subcode. By using the Euclidean algorithm mentioned before, we can find polynomials $P_F(X)$ and $Q_F(X)$ such that

$$P_F(X)h^*(X)I_f(X) + Q_F(X)h^*(X)g^*(X) = h^*(X). \quad (4.28)$$

Comparing Eqs. (4.27) and (4.28), we obtain

$$C_i(X) = P_F(X).I_i'(X),$$
$$D_i(X) = Q_F(X).I_i'(X). \quad (4.29)$$

Alternatively, the tap-coefficient polynomial $C_i(X)$ can be found through the so-called idempotent function [26], [32], [45]. For every cyclic code V,

there exists a unique polynomial E(X) called the idempotent of V such that

1) $E(X) = E^2(X) = E(X^2) \bmod (1+X^n)$;

2) E(X) generates V, and

3) $E(X)V(X)=V(X) \bmod (1+X^n)$ for $V(X) \in V$.

To find the above idempotent function for a dual code V* of the cyclic code V, we recall that h*(X) and g*(X) are respectively the generator and the parity-check polynomials of V* if and only if g(X) and h(X) are respectively the generator and the parity-check polynomials of V. We have $1+X^n=h^*(X)g^*(X)$ for polynomials over GF(2). Since $1+X^n$ has distinct factors, we also have that $(h^*(X),g^*(X))=1$. Hence, by the Euclidean algorithm, there exist polynomials $P_E(X)$ and $Q_E(X)$ such that

$$P_E(X)h^*(X) + Q_E(X)g^*(X) = 1. \qquad (4.30)$$

With respect to the above expression of (4.30), let us define the term

$$E(X) = P_E(X)h^*(X). \qquad (4.31)$$

Clearly, E(X) is a codeword in V*. We multiply (4.30) by $P_E(X)h^*(X)$ so that

$$P_E(X)h^*(X) = P_E^2(X)h^{*2}(X)+P_E(X)Q_E(X)h^*(X)g^*(X)$$
$$= P_E^2(X)h^{*2}(X) \bmod (1+X^n). \qquad (4.32)$$

That is,

$$E(X) = E^2(X) \bmod (1+X^n), \qquad (4.33)$$

so $E(X)=P_E(X)h^*(X)$ is an idempotent function of V*.

When sequential code reduction is applicable, f(X) is a generating orthogonal flat of the dual of the subcode. Consequently, there exist polynomials $P_F(X)$ and $Q_F(X)$ such that

$$P_F(X)f(X) + Q_F(X)(1+X^n) = h^*(X). \qquad (4.34)$$

Multiplying the above by $P_E(X)$ of (4.30), we obtain

$$P(X)f(X) + Q(X)(1+X^n) = E(X) \qquad (4.35)$$

with

$$P(X) = P_E(X).P_F(X);$$
$$Q(X) = P_E(X).Q_F(X). \qquad (4.36)$$

In other words, a polynomial P(X) can be found such that

$$P(X)f(X) = E(X) \bmod (1+X^n), \tag{4.37}$$

where E(X) is the idempotent function of the dual of the subcode.

Since for any codeword $f_i'(X)$ in the dual subcode, the idempotent E(X) has the property that $E(X)f_i'(X) = f_i'(X) \bmod (1+X^n)$, every $f_i'(X)$ can be multiplied by P(X) of (4.35) to obtain a set of $C_i(X)$ which could be used to implement sequential code reduction. Precisely, multiplying both sides of (4.35) by $f_i'(X)$, we get the expression

$$P(X)f_i'(X)f(X) + Q(X)f_i'(X)(1+X^n) = E(X)f_i'(X). \tag{4.38}$$

This is exactly the equation (4.19):

$$C_i(X)f(X) + D_i(X)(1+X^n) = f_i'(X),$$

with

$$C_i(X) = P(X).f_i'(X),$$
$$D_i(X) = Q(X).f_i'(X). \tag{4.39}$$

4.4 EXAMPLES FOR (r,2)TH- PG CODES OF LENGTH 85

In this section, we will give examples on finding orthogonal parity-check sums for majority-logic decoding projective geometry codes. The codes chosen for our demonstration here are the (r,2)th-order binary PG codes of length 85, with r being 2 and 1.

4.4.1 CONVENTIONAL ORTHOGONAL PARITY-CHECK SUMS

As depicted before in Table 2.1, the binary primitive polynomial $P(X)= 1+X^2+X^3+X^4+X^8$ has two primitive factors over $GF(4)= \{0,1,\beta,\beta^2\}$. By taking the first factor $P_1(X)= X^4 +X^3 +\beta X^2 +\beta X +\beta$, we have the corresponding logic

$$\alpha^4 = \alpha^3 + \beta\alpha^2 + \beta\alpha + \beta. \qquad (4.40)$$

From this logic, together with the logic $\beta^2=1+\beta$, the points (α^0), (α^1), ..., (α^{84}) of $PG(3,2^2)$ can be generated. Table 4.1 gives these 0-flats of the geometry $PG(3,2^2)$. It is based on this table we construct the 1- and 2-flats in $PG(3,2^2)$ and accordingly form orthogonal check sums to decode the two-step majority-logic decodable (85,68) binary PG code.

TABLE 4.1 *

Points of Projective Geometry $PG(3,2^2)$

0	0	0	0	1	43	1	2	0	0
1	0	0	1	0	44	3	2	2	2
2	0	1	0	0	45	1	3	3	1
3	1	0	0	0	46	2	1	3	2
4	1	2	2	2	47	3	0	1	3
5	3	0	0	2	48	3	0	2	1
6	3	1	3	1	49	3	3	0	1
7	2	2	0	1	50	0	1	0	1
8	0	3	2	3	51	1	0	1	0
9	3	2	3	0	52	1	3	2	2
10	1	2	1	1	53	2	0	0	2
11	3	3	3	2	54	2	3	1	3
12	0	2	3	1	55	1	2	0	3
13	2	3	1	0	56	3	2	1	2
14	1	2	3	3	57	1	0	3	1
15	3	1	1	2	58	1	1	3	2
16	2	0	3	1	59	0	1	0	2
17	2	0	2	3	60	1	0	2	0
18	2	1	0	3	61	1	0	2	2
19	3	3	0	3	62	1	0	0	2
20	0	1	2	1	63	1	2	0	2
21	1	2	1	0	64	3	2	0	2
22	3	3	2	2	65	1	1	3	1
23	0	3	3	1	66	0	1	3	2
24	3	3	1	0	67	1	3	2	0
25	0	0	1	1	68	2	0	2	2
26	0	1	1	0	69	2	1	1	3
27	1	1	0	0	70	3	2	0	3
28	0	2	2	2	71	1	1	2	1
29	2	2	2	0	72	0	0	3	2
30	0	1	3	3	73	0	3	2	0
31	1	3	3	0	74	3	2	0	0
32	2	1	2	2	75	1	1	1	1
33	3	1	1	3	76	0	3	3	2
34	2	0	2	1	77	3	3	2	0
35	2	1	2	3	78	0	3	1	1
36	3	1	0	3	79	3	1	1	0
37	2	1	2	1	80	2	0	1	1
38	3	1	2	3	81	2	2	2	3
39	2	3	2	1	82	0	1	0	3
40	1	1	2	3	83	1	0	3	0
41	0	0	1	2	84	1	1	2	2
42	0	1	2	0					

* In this table, the first column represents powers of element α; in the remaining columns, the numbers 2 and 3 represent elements β and β^2, respectively.

Consider the line passing through (α^1) and (α^2) which consists of points of the form $(a_1\alpha^1+a_2\alpha^2)$, with $a_1,a_2 \in GF(4)$ and not both zero. These points are the one-dimensional projective subspace of the two-dimensional linear subspace of $GF(4^4)$ over GF(4). From Table 4.1, we see that these points are those field elements $\alpha^i = p_{i3}\alpha^3 + p_{i2}\alpha^2 + p_{i1}\alpha^1 + p_{i0}$ having coefficients $p_{i0}=p_{i3}=0$. That is, they are

$$(\alpha^1),\ (\alpha^2),\ (\alpha^{26}),\ (\alpha^{42}),\ (\alpha^{73}). \qquad (4.41)$$

Each of these points can then connect the point (α^0) to form five different lines intersecting on it.

In particular, the line passing through (α^0) and (α^1) is

$$F_1' = \{(\alpha^0),(\alpha^1),(\alpha^{25}),(\alpha^{41}),(\alpha^{72})\}; \qquad (4.42a)$$

the line passing through (α^0) and (α^2) is

$$F_2' = \{(\alpha^0),(\alpha^2),(\alpha^{50}),(\alpha^{59}),(\alpha^{82})\}; \qquad (4.42b)$$

and the remaining three lines passing through (α^0) and either one of the points (α^{26}), (α^{42}) and (α^{73}) are respectively

$$F_3' = \{(\alpha^0),(\alpha^{26}),(\alpha^{23}),(\alpha^{28}),(\alpha^{76})\}, \qquad (4.42c)$$

$$F_4' = \{(\alpha^0),(\alpha^{42}),(\alpha^{12}),(\alpha^{20}),(\alpha^{78})\}, \qquad (4.42d)$$

and

$$F_5' = \{(\alpha^0),(\alpha^{73}),(\alpha^{8}),(\alpha^{30}),(\alpha^{66})\}. \qquad (4.42e)$$

We have therefore constructed five 1-flats which are orthogonal on the 0-flat (α^0).

Once the five 1-flats which have (α^0) as the common 0-flat are determined, our next task is to form five 2-flats intersecting on each of these 1-flats. A 2-flat that contains the 1-flat $\{(a_0\alpha^0+a_1\alpha^i)\}$ is of the form $((a_0\alpha^0+a_1\alpha^i)+a_2\alpha^j)$, where (α^j) is not in $\{(a_0\alpha^0+a_1\alpha^i)\}$. The 1-flat F_1' given before consists of points $(a_0\alpha^0+a_1\alpha^1)$. The points (α^2) and (α^3) are not in F_1' and thus form a line consisting of the points

$$(\alpha^2),\ (\alpha^3),\ (\alpha^2+\alpha^3)=(\alpha^{27}),$$
$$(\alpha^2+\beta\alpha^3)=(\alpha^{43}),\ (\alpha^2+\beta^2\alpha^3)=(\alpha^{74}). \qquad (4.43)$$

By arranging the above points in different 2-flats, we can connect F_1' to each of them and form five orthogonal 2-flats.

For the 2-flat passing through F_1' and (α^2), we follow the previous discussions on F_1' to F_5' and will get immediately the following:

$$\begin{aligned} F_{11}'' = \{&(\alpha^0);\ (\alpha^1),(\alpha^{25}),(\alpha^{41}),(\alpha^{72});\\ &(\alpha^2),(\alpha^{50}),(\alpha^{59}),(\alpha^{82});\\ &(\alpha^{26}),(\alpha^{23}),(\alpha^{28}),(\alpha^{76});\\ &(\alpha^{42}),(\alpha^{12}),(\alpha^{20}),(\alpha^{78});\\ &(\alpha^{73}),(\alpha^{8}),(\alpha^{30}),(\alpha^{66})\}. \end{aligned} \tag{4.44}$$

Similarly, for forming the 2-flat passing through F_1' and (α^3), we first locate those points on the line connecting (α^1) and (α^3); they are

$$\begin{aligned} &(\alpha^1),\ (\alpha^3),\ (\alpha^1+\alpha^3)=(\alpha^{51}),\\ &(\alpha^1+\beta\alpha^3)=(\alpha^{60}),\ (\alpha^1+\beta^2\alpha^3)=(\alpha^{83}); \end{aligned} \tag{4.45}$$

then we connect (α^0) to each of these points and complete the following 2-flat:

$$\begin{aligned} F_{12}'' = \{&(\alpha^0);\ (\alpha^1),(\alpha^{25}),(\alpha^{41}),(\alpha^{72});\\ &(\alpha^3),(\alpha^5),(\alpha^{53}),(\alpha^{62});\\ &(\alpha^{51}),(\alpha^{17}),(\alpha^{34}),(\alpha^{68});\\ &(\alpha^{60}),(\alpha^{16}),(\alpha^{47}),(\alpha^{61});\\ &(\alpha^{83}),(\alpha^{48}),(\alpha^{57}),(\alpha^{80})\}. \end{aligned} \tag{4.46}$$

In the above manner, the 2-flats passing through F_1' and the respective points of (α^{27}), (α^{43}) and (α^{74}) can all be similarly derived. Table 4.2a indicates these five 2-flats. Note that, for easy references, we do not put the points of the 2-flats into their numerical order. Instead, we intentionally gather those points (with the exception of the common 0-flat {0}) of 1-flats into units.

TABLE 4.2a*

Polynomials orthogonal on {0, 1,25,41,72}

{0, 1,25,41,72; 2,50,59,82;26,23,28,76;
42,12,20,78;73, 8,30,66},

{0, 1,25,41,72; 3, 5,53,62;51,17,34,68;
60,16,47,61;83,48,57,80},

{0, 1,25,41,72;27, 7,19,49;24,40,71,84;
29,11,75,81;77,22,58,65},

{0, 1,25,41,72;43,36,55,63;13,14,38,54;
21, 6,10,39;79, 4,15,33},

{0, 1,25,41,72;74,18,64,70; 9,32,35,37;
31,44,45,69;67,46,52,56}.

* In this and the following tables, the integers inside the brackets are powers of X.

TABLE 4.2b

Polynomials orthogonal on {0, 2,50,59,82}

```
{0, 2,50,59,82; 1,25,41,72;26,23,28,76;
                42,12,20,78;73, 8,30,66},

{0, 2,50,59,82; 3, 5,53,62;27, 7,19,49;
                43,36,55,63;74,18,64,70},

{0, 2,50,59,82;51,17,34,68; 9,32,35,37;
                21, 6,10,39;29,11,75,81},

{0, 2,50,59,82;60,16,47,61;24,40,71,84;
                67,46,52,56;79, 4,15,33},

{0, 2,50,59,82;83,48,57,80;13,14,38,54;
                31,44,45,69;77,22,58,65}.
```

TABLE 4.2c

Polynomials orthogonal on {0,26,23,28,76}

```
{0,26,23,28,76; 1,25,41,72; 2,50,59,82;
                42,12,20,78;73, 8,30,66},

{0,26,23,28,76; 3, 5,53,62;29,11,75,81;
                31,44,45,69;79, 4,15,33},

{0,26,23,28,76;51,17,34,68;13,14,38,54;
                27, 7,19,49;67,46,52,56},

{0,26,23,28,76;60,16,47,61; 9,32,35,37;
                43,36,55,63;77,22,58,65},

{0,26,23,28,76;83,48,57,80;21, 6,10,39;
                24,40,71,84;74,18,64,70}.
```

TABLE 4.2d

Polynomials orthogonal on {0,42,12,20,78}

```
{0,42,12,20,78; 1,25,41,72; 2,50,59,82;
                26,23,28,76;73, 8,30,66},

{0,42,12,20,78; 3, 5,53,62; 9,32,35,37;
                13,14,38,54;24,40,71,84},

{0,42,12,20,78;51,17,34,68;74,18,64,70;
                77,22,58,65;79, 4,15,33},

{0,42,12,20,78;60,16,47,61;21, 6,10,39;
                27, 7,19,49;31,44,45,69},

{0,42,12,20,78;83,48,57,80;29,11,75,81;
                43,36,55,63;67,46,52,56}.
```

TABLE 4.2e

Polynomials orthogonal on {0,73, 8,30,66}

```
{0,73, 8,30,66; 1,25,41,72; 2,50,59,82;
                26,23,28,76;42,12,20,78},

{0,73, 8,30,66; 3, 5,53,62;21, 6,10,39;
                67,46,52,56;77,22,58,65},

{0,73, 8,30,66;51,17,34,68;24,40,71,84;
                31,44,45,69;43,36,55,63},

{0,73, 8,30,66;60,16,47,61;13,14,38,54;
                29,11,75,81;74,18,64,70},

{0,73, 8,30,66;83,48,57,80; 9,32,35,37;
                27, 7,19,49;79, 4,15,33}.
```

Following the above procedures, for each 1-flat F_i', $1 \leq i \leq 5$, we can form five 2-flats orthogonal on it. The incidence vectors of these 25 2-flats are given in Tables 4.2a to 4.2e. We have therefore constructed the 1- and 2-flats of PG(3,2^2), with five 2-flats orthogonal on an appropriate 1-flat, and with five appropriate 1-flats orthogonal on the specific 0-flat (α^0).

Using the incidence polynomials of the 25 2-flats given in Tables 4.2a to 4.2e, two levels of parity-check sums can be formed to decode the discussed (85,68) binary PG code. Let S_i' denote the sum of error digits at the locations corresponding to the points on F_i' for $1 \leq i \leq 5$. Then for each error sum S_i', there are five parity-check sums S_{i1}'', S_{i2}'', ..., S_{i5}'' orthogonal on it which are formed based on the incidence vectors of the five orthogonal 2-flats. Thus, S_i' can be determined correctly provided that there are two or fewer errors in the error vector. The error sums S_1', S_2', S_3', S_4' and S_5' are orthogonal on the error digit e_0. Consequently, e_0 can be determined from these error sums. Thus, the (85,68) binary PG code is two-step majority-logic decodable. Figure 4.4a shows a two-level majority-logic decoding cirtuit for the above PG code.

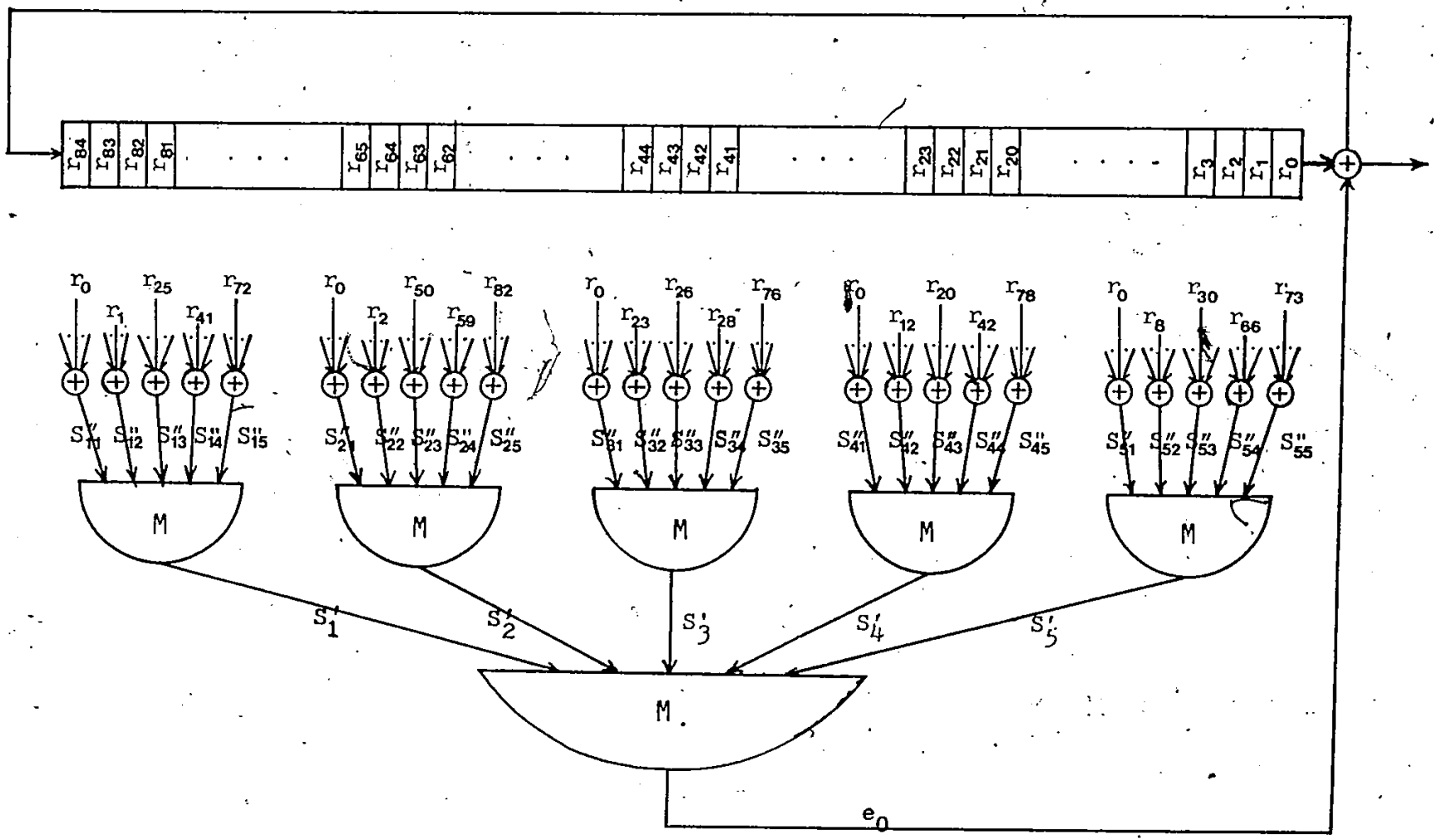


Figure 4.4a Conventional two-level majority-logic decoder for the (85,68) binary PG code.

As for the (1,2)th-order (85,24) binary PG code, we note that it is 1-step majority-logic decodable and hence has its parity-check space all the incidence vectors of the 1-flats in $PG(3,2^2)$. From the previous discussions on forming 2-flats of $PG(3,2^2)$, we recall that the 2-flats are unions of the appropriate 1-flats in $PG(3,2^2)$. Hence we can simply decompose the 2-flats to get 1-flat components. The collection of these 1-flats will then form a check set for the (1,2)th- PG code of length 85.

For example, on referring Table 4.2a, the 2-flat F''_{11} of (4.44) can be decomposed into the five 1-flats given in (4.42); the 2-flat F''_{12} of (4.46) can be decomposed into the five 1-flats:

$$\{0, 1,25,41,72\},$$
$$\{0, 3, 5,53,62\}, \{0,51,17,34,68\},$$
$$\{0,60,16,47,61\}, \{0,83,48,57,80\}; \qquad (4.47a)$$

the 2-flat F''_{13} of the 3rd row of Table 4.2a can be decomposed into the five 1-flats:

$$\{0, 1,25,41,72\},$$
$$\{0,27, 7,19,49\}, \{0,24,40,71,84\},$$
$$\{0,29,11,75,81\}, \{0,77,22,58,65\}; \qquad (4.47b)$$

and so on for the other 2-flats. We can therefore obtain 21 1-flats orthogonal on the 0-flat {0}. Table 4.3 gives the incidence polynomials for these 21 1-flats. The error digit e_0 can therefore be determined correctly from the parity-check sums formed from the above 1-flats provided that there are 10 or fewer errors in the error vector. Figure 4.4b schematically gives the one-level majority-logic decoder for the (85,24) PG code.

TABLE 4.3

Incidence Polynomials for (1,2)th-order binary PG code of length 85

{0, 1, 25, 41, 72}	{0, 16, 47, 60, 61}
{0, 2, 50, 59, 82}	{0, 17, 34, 51, 68}
{0, 3, 5, 53, 62}	{0, 18, 64, 70, 74}
{0, 4, 15, 33, 79}	{0, 22, 58, 65, 77}
{0, 6, 10, 21, 39}	{0, 23, 26, 28, 76}
{0, 7, 19, 27, 49}	{0, 24, 40, 71, 84}
{0, 8, 30, 66, 73}	{0, 31, 44, 45, 69}
{0, 9, 32, 35, 37}	{0, 36, 43, 55, 63}
{0, 11, 29, 75, 81}	{0, 46, 52, 56, 67}
{0, 12, 20, 42, 78}	{0, 48, 57, 80, 83}
{0, 13, 14, 38, 54}	

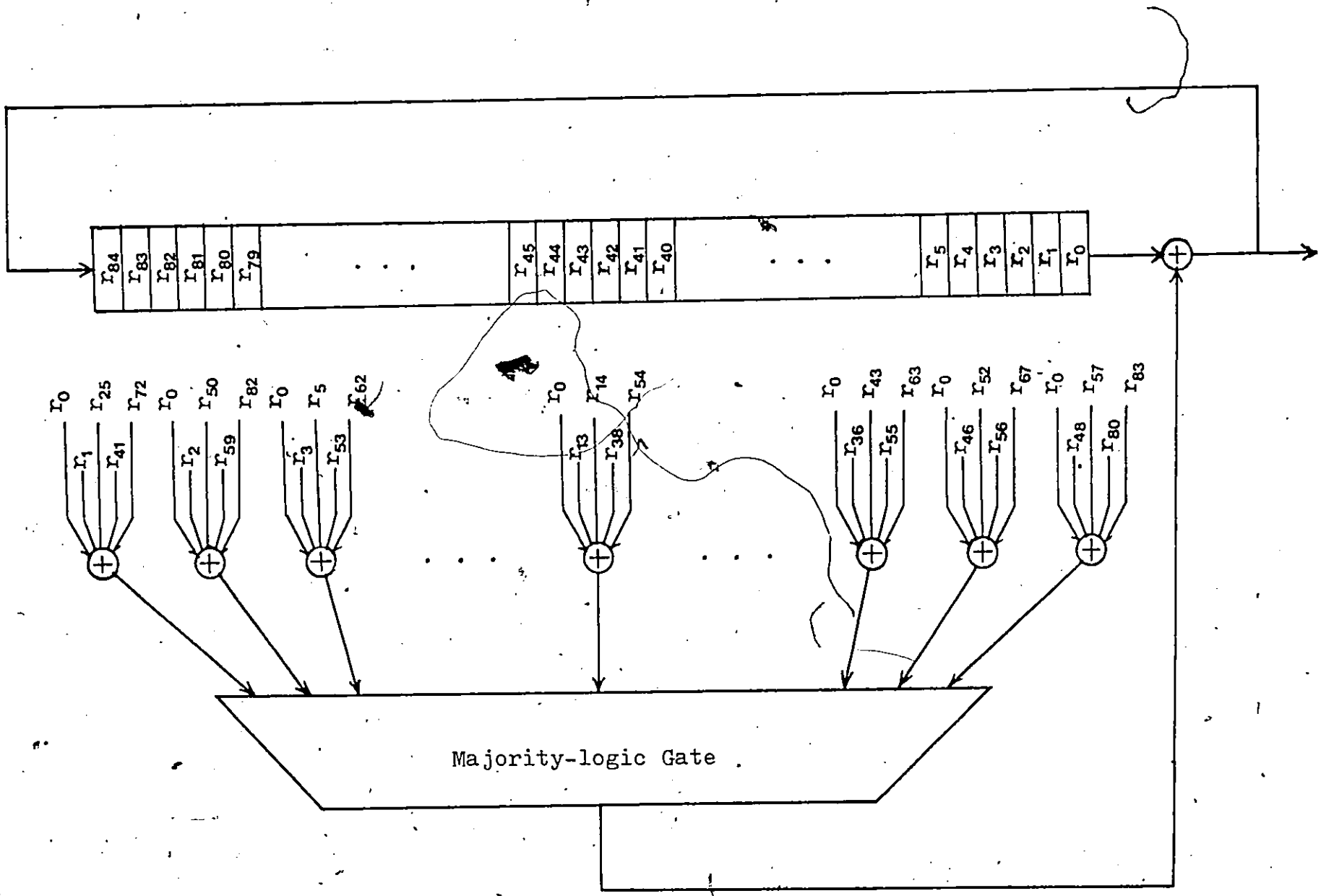


Figure 4.4b The one-level majority-logic decoder for the (85,24) binary PG code.

4.4.2 CHECK SETS FOR SEQUENTIAL-CODE-REDUCTION DECODING

So far, the conventional rule of orthogonalization has been used for decoding the (2,2)th-order binary (85,68) PG code. It is seen that two levels of orthogonalization and a total of six five-input majority-logic gates are needed. However, if we examine the geometry $PG(3,2^2)$ carefully, the code can be decoded in two steps using sequential-code-reduction majority-logic decoding.

Consider the five 1-flats F_1' to F_5' given in (4.42a)--(4.42e). The incidence polynomials corresponding to these 1-flats are:

$$\begin{aligned} f_1'(X) &= 1 + X + X^{25} + X^{41} + X^{72}, \\ f_2'(X) &= 1 + X^2 + X^{50} + X^{59} + X^{82}, \\ f_3'(X) &= 1 + X^{23} + X^{26} + X^{28} + X^{76}, \\ f_4'(X) &= 1 + X^{12} + X^{20} + X^{42} + X^{78}, \\ f_5'(X) &= 1 + X^8 + X^{30} + X^{66} + X^{73}. \end{aligned} \tag{4.48}$$

These 1-flats are orthogonal on the 0-flat $f_0(X)=1$. Furthermore, for each of the above 1-flats, there exist five 2-flats orthogonal on it.

In order to form parity-check sums for sequential-code-reduction majority-logic decoding, we have to find a flat among $f_1'(X)$ to $f_5'(X)$ such that this flat divides all other flats, in the sense that the manipulations are taken modulo $1+X^{85}$. The condition for the existence of such a generating orthogonal flat $f(X)$ is that $GCD(f(X),1+X^{85})$ is the reciprocal of the parity-check polynomial of the (1,2)th-order binary PG code of length 85. Equivalently, it is the generator polynomial of the dual of the (85,24) PG subcode.

By taking $f_1'(X)= 1+X+X^{25}+X^{41}+X^{72}$, and with $g_1^*(X)$ being the reciprocal of (3.48), we find that

$$\begin{aligned} &GCD\ (f_1'(X),\ 1+X^{85}) \\ &= 1+X^2+X^3+X^4+X^7+X^{10}+X^{11} \\ &\quad +X^{12}+X^{14}+X^{15}+X^{17}+X^{19}+X^{24} \\ &= (1+X^{85})/g_1^*(X) = h_1^*(X). \end{aligned} \tag{4.49}$$

This is exactly the generator polynomial of the dual of the (1,2)th-order binary (85,24) PG code. Hence, by letting $f(X)=f_1'(X)= 1+X+X^{25}+X^{41}+X^{72}$ be the polynomial associated to the first parity-check sum S_1' formed at the output of the first-level majority-logic gate, each incidence polynomial $f_i'(X)$ given in (4.48) can be expressed as a multiple of $f(X)$ mod $(1+X^{85})$.

Specifically, with reference to the generator and parity-check polynomials of the (1,2)th-order binary (85,24) PG code, as given in Eqs. (3.48) and (3.49), we have

$$P_E(X) = 1+X^2+X^{12}+X^{18}+X^{22}+X^{30}+X^{32}+X^{34} \\ +X^{36}+X^{38}+X^{46}+X^{48}+X^{52}+X^{54}+X^{56} \qquad (4.50)$$

to be the polynomial defining the idempotent function $E(X) = P_E(X)h_1^*(X)$ of the dual of the (85,24) PG code. Also, we have

$$P_F(X) = X^5+X^8+X^{10}+X^{14}+X^{15}+X^{17}+X^{19}+X^{22}+X^{23} \\ +X^{25}+X^{28}+X^{29}+X^{31}+X^{33}+X^{35}+X^{38}+X^{39} \\ +X^{41}+X^{44}+X^{45}+X^{47}+X^{49}+X^{51}+X^{52}+X^{54} \qquad (4.51)$$

to be the polynomial defining $P_F(X)f(X) = h_1^*(X) \mod (1+X^{85})$. Hence we have the polynomial

$$P(X) = P_E(X)P_F(X) \mod (1+X^{85}) \\ = 1+X+X^2+X^5+X^6+X^8+X^9+X^{11}+X^{12}+X^{18}+X^{19}+X^{20} \\ +X^{21}+X^{22}+X^{24}+X^{25}+X^{27}+X^{31}+X^{33}+X^{34}+X^{35} \\ +X^{36}+X^{37}+X^{39}+X^{40}+X^{44}+X^{46}+X^{47}+X^{50}+X^{51} \\ +X^{52}+X^{53}+X^{55}+X^{57}+X^{58}+X^{60}+X^{62}+X^{63}+X^{65} \\ +X^{66}+X^{68}+X^{69}+X^{73}+X^{74}+X^{76}+X^{82}+X^{83}. \qquad (4.52)$$

From the above polynomial P(X), together with the 1-flats given in (4.48), the tap-coefficient polynomials satisfying $f_i'(X) = C_i(X)f(X) \bmod (1+X^{85})$ are found to be

$$C_1(X) = P(X)f_1'(X) = 1,$$

$$C_2(X) = P(X)f_2'(X)$$
$$= 1 + X + X^{25} + X^{41} + X^{72},$$

$$C_3(X) = P(X)f_3'(X)$$
$$= X^{13} + X^{26} + X^{27} + X^{51} + X^{67},$$

$$C_4(X)=P(X)f_4'(X)$$
$$=1+X+X^3+X^4+X^6+X^7+X^8+X^{11}+X^{14}+X^{20}+X^{21}$$
$$+X^{22}+X^{24}+X^{25}+X^{26}+X^{30}+X^{32}+X^{35}+X^{37}+X^{38}$$
$$+X^{39}+X^{41}+X^{43}+X^{45}+X^{46}+X^{48}+X^{49}+X^{50}+X^{51}$$
$$+X^{55}+X^{56}+X^{58}+X^{59}+X^{60}+X^{63}+X^{64}+X^{68}$$
$$+X^{71}+X^{73}+X^{76}+X^{77}+X^{80}+X^{82}+X^{83}+X^{84},$$

$$C_5(X)=P(X)f_5'(X)$$
$$=X^2+X^8+X^9+X^{10}+X^{12}+X^{13}+X^{14}+X^{18}+X^{20}+X^{23}$$
$$+X^{25}+X^{26}+X^{27}+X^{29}+X^{31}+X^{33}+X^{34}+X^{36}+X^{37}$$
$$+X^{38}+X^{39}+X^{43}+X^{44}+X^{46}+X^{47}+X^{48}+X^{51}+X^{52}$$
$$+X^{56}+X^{59}+X^{61}+X^{64}+X^{65}+X^{68}+X^{70}+X^{71}+X^{72}$$
$$+X^{73}+X^{74}+X^{76}+X^{77}+X^{79}+X^{80}+X^{81}+X^{84}.$$

(4.53)

Based on the above tap-coefficient polynomials, the two-stage sequential-code-reduction majority-logic decoding circuit for the (85,68) binary PG code is as that shown in Figure 4.5.

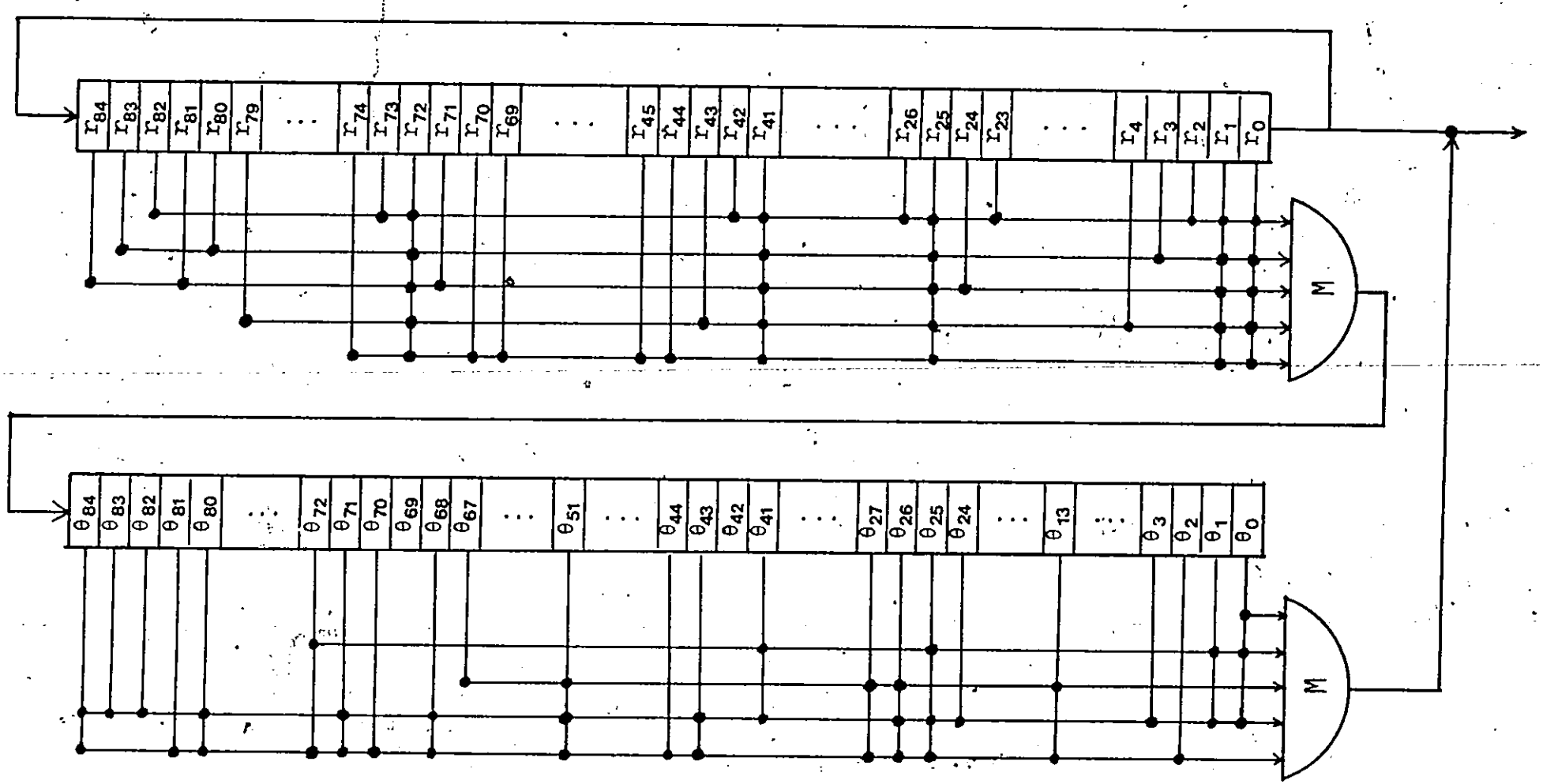


Figure 4.5 The two-stage sequential-code-reduction majority-logic decoder for the (85,68) binary PG code.

Chapter V

CONCLUDING REMARKS

In this thesis, we have examined the highly mathematical structured projective geometry codes. Our main emphasis has been on determining coset representatives of acceptable cyclotomic cosets and on constructing flats in finite projective geometry PG(m,q). The knowledge of coset representatives plays an important role in finding the generator polynomial to encode the code. The knowledge of flats in PG(m,q) plays an important role in forming parity-check sums to decode the code.

On the encoding side, we first quoted the conditions characterizing roots of generator polynomial g(X) for an (r,s)th-order binary projective geometry code of length $n = (q^{m+1}-1)/(q-1)$, $q=2^s$. Based on these conditions, together with the fact that if ν is an acceptable number for $\alpha^{\nu} \in GF(q^{m+1})$ to be a root of g(X), then so is every number belonging to the cyclotomic coset containing ν, we developed two

different approaches on the problem of finding coset representatives.

In the 2's exponent approach depicted in Section 3.1, we utilized all possible 2's exponent combinations to form acceptable integers which characterize roots of generator polynomials. A coset partition is then made on those valid odd numbers and different cyclotomic cosets formed. From each of these cyclotomic cosets, an odd number is selected as the representative.

In the cyclic classes approach discussed in Sections 3.2 and 3.3, on the basis of the concept that the cyclotomic coset containing a positive integer ν is the union of the cyclic class containing ν and the cyclic classes containing $2^{\ell}\nu$'s, $0 \leq \ell < s$, we derived certain sharper results. The derivations involved in this approach are much more complicated, but the results are much easier for quick references.

By using the above lines of attack, we have successfully built up rules for tackling coset representatives for the (m-1,s)th- and the (m-2,2)th-binary PG codes. In particular, we tabulated sharper

results for codes with parameters $m-r=1$ and $s=2,3$, and for codes with $m-r=2$ and $s=2$. From these tabulations, if the representatives are $L_0=0$, L_1, L_2, ..., L_V, then the generator polynomial $g(X)$ is simply the product of the minimal functions of the elements α^{L_i}'s which belong to $GF(q^{m+1})$.

Though our discussions on coset representatives have been restricted to limited classes of PG codes, we feel that the same idea can be similarly extended to other classes of PG codes. For example, with reference to the discussion of Section 3.3, the $(m-2,s)$th-order PG codes can be considered for $q=2^s$ sub-conditions characterizing the radix-q weights $\max_{0\le \ell<s} W_q(2^\ell \nu) \le 2(q-1)$; the $(m-3,s)$th-order PG codes can be considered for $q=2^s$ sub-conditions characterizing the radix-q weights $\max_{0\le \ell<s} W_q(2^\ell \nu) \le 3(q-1)$; and so on for the other PG codes. Obviously, the problem lies in that when s becomes large, the logical conditions to be considered grow enormous. Hence, how to minimize these logical conditions is an important subject for a further research.

On the decoding side, we first justified the correspondences between the projective geometry PG(m,q) and the Galois field $GF(q^{m+1})$. By choosing a binary primitive polynomial of degree s(m+1) and factorizing it methodically, we obtained s irreducible polynomial factors over GF(q), each of degree m+1. Any one of these primitive factors can then be used as a logic to construct points of PG(m,q). These points are also termed 0-flats.

The 0-flats derived above can be expressed as combinations of the basis elements of $GF(q^{m+1})$. This alternative expression is convenient for making calculations over $GF(q^{m+1})$ and hence useful for deriving r-flats in PG(m,q). For any (r-1)-flat and any point not belonging to it, there is a unique r-flat in PG(m,q) which contains both the (r-1)-flat and the point. By recursively applying this configuration of projective geometry for r=1, 2, ..., m-1, we determined the r-flats in PG(m,q) that intersect on a given (r-1)-flat in PG(m,q). The recursion procedures here involved series of searches over the table of 0-flats expressed in combinations of the basis elements of $GF(q^{m+1})$.

From the above recursion algorithm, it is seen that the proper connection between an (r-1)-flat and the appropriate points plays the key role on constructing an r-flat. To find these appropriate points and to allocate points within the r-flat, a scratch pad of memory is required to remember the location of all sub-flats. Apparently, for large values of code lengths and/or large numbers of decoding steps, the scratch pad of storage grows enormously. Though sequential-code-reduction technique can lessen this problem a little, it is still one of the aspects worth of taking a further investigation.

Through sequential code reduction (SCR) technique, a significant reduction in decoder complexity can be obtained at the expense of a modest increase in decoding time. In general, for PG codes with large r and J, the SCR majority-logic decoding technique can greatly reduce the number of majority-logic gates. Furthermore, the number of modulo-2 adders needed to form check sums are also reduced significantly. There is an increase of buffer storage, from one buffer register of n stages (for storing the received vector) to r buffer registers. The major disadvantage of SCR majority-logic decoding is

the increase in decoding time. The conventional majority-logic decoding requires n clock times to decode the received vector; however, the SCR majority-logic decoding requires nr clock times to decode the received vector.

The difficulty with SCR majority-logic decoding is that it is hard to find the generating orthogonal flat at each level of orthogonalization. Also, it is still not yet known whether the generating orthogonal flats exist for the whole class of PG codes. Only through computer verifications, it is shown that all projective geometry codes of length $n < 5461$ can be decoded by an r-stage sequential code reduction algorithm using J orthogonal check sums and one majority-logic gate at each stage. Though we are unable to prove whether this holds for all PG codes, we conjecture that any r-step orthogonalizable PG code can be decoded up to $d=J+1$ by an SCR majority-logic decoding algorithm. If this conjecture proves to hold, then it becomes possible to consider the use of very long r-step orthogonalizable PG codes in practical error-control systems.

Appendix A

GENERATOR POLYNOMIALS OF CERTAIN BINARY PG CODES

The generator polynomials of certain ~~(r,s)th~~-order binary projective geometry codes of length $n=(q^{m+1}-1)/(q-1)$ are given here, where $q=2^s$. They are given in octal representations. Each digit in the representation is coded as follows:

0 — 000, 1 — 001, 2 — 010, 3 — 011,
4 — 100, 5 — 101, 6 — 110, 7 — 111.

When the octal representation is expanded in binary, the binary digits are the coefficients of the polynomial, with the high-order coefficients at the left for the regular polynomial (.), and with the high-order coefficients at the right for the reciprocal polynomial (.)*.

m+1 = 3, r = 1, s = 2:

$$g(X) = (003)\ .(127)\ .(015)$$
$$= 2325$$

m+1 = 4, r = 2, s = 2:

$$g1(X) = (003)\ .(567)\ .(675)$$
$$= 407005$$

m+1 = 4, r = 1, s = 2:

$$g2(X) = g1(X)\ .(727)\ .(613)\ .(477)$$
$$.(477)*.(471)\ .(037)$$
$$= 260230230235241131241$$

m+1 = 5, r = 3, s = 2:

g1(X) = (0003) .(2017) .(2257) .(0075)

= 467017713

m+1 = 5, r = 2, s = 2:

g2(X) = g1(X) .(2653) .(3753) .(3573) .(2107)
.(3061) .(2547) .(3121) .(2701) .(2437)
.(0067) .(3607) .(2355) .(0051)
= 515375202531157535371315137777465676140100073310 3

m+1 = 5, r = 1, s = 2:

g3(X) = g2(X) .(2653)*.(2311) .(3777) .(3121)*
.(2311)*.(2107)*.(3573)*.(3315) .(3607)*
.(3061)*.(2547)*.(3315)*.(2355)*.(0067)*
.(2437)*.(0051)*
= 767337675032376750444451426557567023620275020767341146032575047570122224767341171014756743704033641

m+1 = 6, r = 4, s = 2:

g1(X) = (00003) .(12133) .(11765) .(13311)

= 3062010324541

m+1 = 6, r = 3, s = 2:

g2(X) = g1(X) .(13077) .(10065) .(14405) .(13321)
.(15173) .(10355) .(12331) .(16663) .(16267)
.(17545) .(17323) .(13617) .(10571) .(13571)
.(17657) .(13517) .(10621) .(16017) .(00165)
.(14067) .(00023) .(15775) .(17073) .(00013)

= 72715727611310132207211135700254544347750622461450343325362761132752550727630613037751666240007005317

m+1 = 6, r = 2, s = 2:

g3(X) = g2(X) .(10761) .(17361) .(14755) .(15415)
.(16475) .(13475) .(13475)*.(10213) .(14111)
.(13535) .(10603) .(16457) .(11657) .(10167)
.(13737) .(10603)*.(15415)*.(16137) .(17141)
.(13737)*.(11763) .(17777) .(17233) .(11703)
.(16003) .(14537) .(15347) .(11045) .(10167)*
.(13003) .(10077) .(14313) .(11073) .(10743)
.(15353) .(14313)*.(14111)*.(11657)*.(12265)
.(14411) .(10743)*.(14411)*.(10077)*.(11073)*
.(11045)*.(12345) .(16571) .(16571)*.(00037)

m+1 = 6, r = 1, s = 2:

g4(X) = g3(X) .(13077)*.(16475)*.(17141)*.(14755)*
.(12331)*.(17233)*.(16137)*.(10571)*.(17361)*
.(13321)*.(14405)*.(16457)*.(11763)*.(13617)*
.(11703)*.(13571)*.(15173)*.(10355)*.(13003)*
.(15347)*.(14067)*.(14537)*.(13517)*.(17323)*
.(16003)*.(15775)*.(16017)*.(00165)*.(10213)*
.(17657)*.(16267)*.(00007) .(12265)*.(10621)*
.(17073)*.(00013)*

m+1 = 7, r = 5, s = 2:

g1(X) = (00003) .(40547) .(54055) .(52737) .(00203)

= 67652252233647405

m+1 = 7, r = 4, s = 2:

g2(X) = g1(X) .(47645) .(46425) .(62603) .(62115)
.(64715) .(67653) .(74247) .(72501) .(73757)
.(60523) .(56733) .(57745) .(46047) .(47727)
.(76175) .(42711) .(54175) .(51275) .(47043)
.(62431) .(53747) .(53043) .(66031) .(54473)
.(50343) .(65645) .(00253) .(71323) .(43723)
.(55247) .(43377) .(62055) .(41551) .(41171)
.(74117) .(00217) .(77211) .(00271)

= 151521310200354764246225375431700754354264
476767370212652500527155371224724666313521
211436771042312402324370232533316012763447
716475146474324221111151243123433040372175
33151753166403572161

m+1 = 8, r = 6, s = 2:

g1(X) = (000003) .(215435) .(225657)
.(310327) .(357047)
= 624305437306365016473l

m+1 = 8, r = 5, s = 2:

g2(X) = g1(X) .(311513) .(363501) .(357333) .(327721)
.(374111) .(237403) .(314061) .(323527) .(237421)
.(261105) .(351641) .(365705) .(315633) .(230535)
.(371771) .(301663) .(245351) .(362053) .(347211)
.(362555) .(317171) .(344733) .(273211) .(207753)
.(271725) .(271251) .(200365) .(254241) .(227157)
.(341343) .(241245) .(277215) .(311661) .(205003)
.(277745) .(371643) .(322367) .(273007) .(261177)
.(317531) .(223561) .(222123) .(000573) .(223463)
.(310267) .(343055) .(307713) .(243043) .(253207)
.(222541) .(333515) .(254651) .(331173) .(361707)
.(323157) .(351701) .(000771)

m+1 = 3, r = 1, s = 3:

g(X) = (0003) .(1231) .(1027) .(1401)

= 2120014741

m+1 = 4, r = 2, s = 3:

g1(X) = (00003) .(12153) .(10065) .(10377)
.(17703) .(11433) .(00023)

= 5161070255137170012505

m+1 = 4, r = 1, s = 3:

g2(X) = g1(X) .(10761) .(15115) .(10011) .(13617)
.(14315) .(17657) .(16757) .(13535) .(13157)
.(13563) .(10167) .(16757)*.(14315)*.(12315)
.(17777) .(11637) .(12673) .(15347) .(14043)
.(10167)*.(14043)*.(10245) .(15353) .(00111)
.(12673)*.(13413) .(10245)*.(12345) .(00037)
.(00007)

= 54670445276350614143302625067741340664677177056721206762626476045417156676353607747546556705745606540236400332762525605003702411126571

m+1 = 5, r = 3, s = 3:

g(X) = (000003) .(125253) .(127701) .(137733) .(125263)
.(122707) .(103751) .(136321) .(121125) .(000057)
= 1563714747622037320437411155036673325442401

m+1 = 3, r = 1, s = 4:

g(X) = (00003) .(13077) .(15173) .(16267) .(13617)
.(10571) .(10621) .(00165) .(00013)
= 2425002105200666750745212005

m+1 = 4, r = 2, s = 4:

g(X) = (000003) .(311513) .(374111) .(237421) .(365705)
.(315633) .(245351) .(347211) .(317171) .(277215)
.(273007) .(317531) .(222123) .(243043) .(253207)
.(333515) .(331173)
= 55076601332304727215255655103243371442777204652436774717206553252623724655200020252425

m+1 = 3, r = 1, s = 5:

g(x) = (000003) .(103437) .(124677) .(150451) .(142327)
.(114533) .(134105) .(134411) .(121437) .(115313)
.(163761) .(166761) .(147553) .(104073) .(160335)
.(121055) .(100725) .(000013)
= 33404461470652526310440567124100645322343 7
22626306734547503777265637417520252124 05

Appendix B

PARTIAL LIST OF PARITY-CHECK SETS OF (m-1,s)TH- PG CODES

For the first several (r=m-1,s)th-order binary PG codes of length $n= (q^{m+1}-1)/(q-1)$, $q=2^s$, we list here one of the orthogonal parity-check sets used at each μ-th decoding step, $1 \le \mu \le r$. Note that, for want of space, when $\mu \ge 2$, the common $(\mu-1)$-flat to which the $q+1$ μ-flats are orthogonal with is simply put on the top of the check set.

The numbers listed are the indexes corresponding to the bit positions within the code block. For easy references, these position indexes are not put into their numerical order. Instead, they are intentionally gathered in units of sub-flats. With this arrangement, it is easy to successively decompose the given check sums into those of the $(m-\mu,s)$th-order PG codes of the same block length n.

$m+1 = 3,\ s = 2,\ r = 1$:

0	1	6	8	18,
0	2	12	15	16,
0	7	5	17	20,
0	9	3	4	11,
0	19	10	13	14.

$m+1 = 4,\ s = 2,\ r = 2$:

0	1	25	41	72,
0	2	50	59	82,
0	26	23	28	76,
0	42	12	20	78,
0	73	8	30	66.

0 1 25 41 72

2	50	59	82	26	23	28	76
42	12	20	78	73	8	30	66,
3	5	53	62	51	17	34	68
60	16	47	61	83	48	57	80,
27	7	19	49	24	40	71	84
29	11	75	81	77	22	58	65,
43	36	55	63	13	14	38	54
21	6	10	39	79	4	15	33,
74	18	64	70	9	32	35	37
31	44	45	69	67	46	52	56.

m+1 = 5, s = 2, r = 3:

0	1	32	77	242,
0	2	64	143	154,
0	33	37	161	319,
0	78	49	151	274,
0	243	58	109	204.

0	1	32	77	242			
2	64	143	154	33	37	161	319
78	49	151	274	243	58	109	204,
3	10	92	222	65	126	295	300
144	69	228	271	155	105	193	325,
34	52	233	281	38	170	186	291
162	135	147	175	320	96	216	284,
79	90	277	279	50	88	220	236
152	62	141	339	275	8	231	256,
244	81	238	258	59	22	55	183
110	25	44	118	205	72	114	306.

0	1	32	77	242			
2	64	143	154	33	37	161	319
78	49	151	274	243	58	109	204
3	10	92	222	65	126	295	300
144	69	228	271	155	105	193	325
34	52	233	281	38	170	186	291
162	135	147	175	320	96	216	284
79	90	277	279	50	88	220	236
152	62	141	339	275	8	231	256
244	81	238	258	59	22	55	183
110	25	44	118	205	72	114	306,

4	128	286	308	11	198	200	262
93	19	85	316	223	248	267	333
66	74	297	322	127	84	197	266
296	165	264	265	301	166	313	328
145	67	116	218	70	139	214	298
229	181	289	323	272	75	159	202
156	98	207	302	106	41	167	336
194	15	28	329	326	13	179	314,
35	107	149	240	53	168	191	254
234	42	133	269	282	124	304	337
39	137	195	246	171	16	121	209
187	189	251	330	292	29	102	225
163	157	177	260	136	101	208	250
148	132	253	303	176	99	100	131
321	83	164	327	97	14	178	335
217	158	180	213	285	261	315	332,
80	24	54	71	91	192	227	299
278	87	140	255	280	169	174	215
51	146	185	283	89	7	219	338
221	68	104	125	237	21	117	305
153	36	57	273	63	150	203	318
142	48	108	160	340	31	76	241
276	61	230	235	9	270	294	324
232	95	134	290	257	43	113	182,
245	120	188	224	82	212	331	334
239	123	190	268	259	130	249	252
60	94	112	293	23	86	173	226
56	30	47	317	184	6	20	103
111	5	46	172	26	17	287	311
45	210	309	310	119	122	129	211
206	12	27	40	73	196	263	312
115	138	201	288	307	18	199	247.

m+1 = 6, s = 2, r = 4:

0 3 724 877 1080,

0 1 448 794 1000,

0 224 397 500 683,

0 636 381 621 1183,

0 897 649 800 907.

0 3 724 877 1080

1 448 794 1000 224 397 500 683
636 381 621 1183 897 649 800 907,

2 223 635 896 450 24 191 332
796 355 933 1281 1002 55 454 1029,

449 235 429 1298 7 64 311 1210
430 199 1257 1344 1273 170 837 1223,

795 6 83 389 84 439 880 1017
236 701 749 1083 763 531 592 727,

1001 762 1242 1272 390 325 585 1235
1243 181 270 342 1299 671 1006 1045.

0 3 724 877 1080
1 448 794 1000 224 397 500 683
636 381 621 1183 897 649 800 907

2 223 635 896 450 24 191 332
796 355 933 1281 1002 55 454 1029
449 235 429 1298 7 64 311 1210
430 199 1257 1344 1273 170 837 1223
795 6 83 389 84 439 880 1017
236 701 749 1083 763 531 592 727
1001 762 1242 1272 390 325 585 1235
1243 181 270 342 1299 671 1006 1045,

```
   4  427  446 1270  451   21  220 1278
 797  618  680  997 1003  178  322  759
 725  647 1233 1342   25   71  156 1172
 356  153  641  644 1030   39  293  359
 878  168  395  669   56  125  512 1123
 333  467  937 1325 1282  306  712 1288
1081  309  340 1181  192  163  870 1291
 455  130  520  715  934  509  666 1285,

 225   12  166  778   85 1101 1294 1319
1274  318  873  993 1300  195  845  975
 398  860 1149 1323  171  135  773 1304
 672  215  782 1311  881  112  250 1024
 501  227  710 1197  838   32  605  686
1007  400 1131 1136 1018  503 1205 1332
 684  362  540 1121  440   42  902 1191
1046  296  948 1064 1224  118 1033 1057,

 637  159 1119 1179    8  854  892 1175
 237   74  266  944 1244   28  256  745
 382   48  664  900   65  260  910 1040
 182  563  803  818  750  652  768 1069
 622   14  128 1055  271  241  364 1126
 702  515  829  862 1211   59  612 1199
1184   89  161 1062  312   50  142  979
 343   16  419  985 1084   91  409  964,

 898  470  858 1231  391  336  790 1338
 431  351  940 1097  764   96  435 1328
 650  780 1105 1170  728  482  542  887
1236  229  629 1360 1345  849  916 1151
 801   37  133  472  532  148  474  523
 586   78  695  718 1258  458 1107 1355
 908  110  639  693  200  568 1186 1248
 326  375  384 1217  593  100  284  624.
```

0	3	724	877	1080			
	1	448	794	1000	224	397	500	683
636	381	621	1183	897	649	800	907
	2	223	635	896	450	24	191	332
796	355	933	1281	1002	55	454	1029
449	235	429	1298	7	64	311	1210
430	199	1257	1344	1273	170	837	1223
795	6	83	389	84	439	880	1017
236	701	749	1083	763	531	592	727
1001	762	1242	1272	390	325	585	1235
1243	181	270	342	1299	671	1006	1045

4	427	446	1270	451	21	220	1278
797	618	680	997	1003	178	322	759
725	647	1233	1342	25	71	156	1172
356	153	641	644	1030	39	293	359
878	168	395	669	56	125	512	1123
333	467	937	1325	1282	306	712	1288
1081	309	340	1181	192	163	870	1291
455	130	520	715	934	509	666	1285
225	12	166	778	85	1101	1294	1319
1274	318	873	993	1300	195	845	975
398	860	1149	1323	171	135	773	1304
672	215	782	1311	881	112	250	1024
501	227	710	1197	838	32	605	686
1007	400	1181	1136	1018	503	1205	1332
684	362	540	1121	440	42	902	1191
1046	296	948	1064	1224	118	1033	1057
637	159	1119	1179	8	854	892	1175
237	74	266	944	1244	28	256	745
382	48	664	900	65	260	910	1040
182	563	803	818	750	652	768	1069
622	14	128	1055	271	241	364	1126
702	515	829	862	1211	59	612	1199
1184	89	161	1062	312	50	142	979
343	16	419	985	1084	91	409	964
898	470	858	1231	391	336	790	1338
431	351	940	1097	764	96	435	1328
650	780	1105	1170	728	482	542	887
1236	229	629	1360	1345	849	916	1151
801	37	133	472	532	148	474	523
586	78	695	718	1258	458	1107	1355
908	110	639	693	200	568	1186	1248
326	375	384	1217	593	100	284	624,

5	234	634	1241	226	539	777	1148
638	36	1169	1230	899	1054	1061	1118
428	388	761	895	160	127	663	1178
779	109	132	857	859	11	361	1196
447	793	999	1364	167	308	426	1341
1180	394	445	646	1232	339	668	1269
1271	82	222	1297	13	47	88	158
471	469	692	1104	1120	165	709	1322
726	54	269	836	383	77	789	1359
399	947	974	1310	802	240	255	984
648	499	620	876	38	321	511	519
665	305	643	996	1324	70	1277	1290
1234	1016	1209	1280	473	283	481	1327
861	73	963	1068	901	1023	1204	1293
1343	331	700	1005	49	58	891	1039
134	604	992	1032	1150	457	567	1096
879	63	354	584	502	249	1100	1190
623	147	434	886	651	408	828	943
169	180	530	1028	15	562	744	1125
228	717	1216	1337	781	295	844	1135
396	380	723	799	129	124	358	758
711	152	679	1284	1171	20	869	936
670	23	748	1256	1056	31	772	872
1106	915	939	1247	1198	141	259	1174
1082	190	198	1044	685	117	317	1303
909	611	853	978	1185	350	848	1354
310	324	438	932	90	265	514	767
111	41	1318	1331	541	95	99	522
341	453	591	1222	363	27	418	817
694	335	374	628	1063	194	214	1130
1182	682	906	1079	162	155	219	466
640	508	617	1287	1122	177	292	714
452	189	931	1078	238	609	826	1052
432	348	787	1228	1301	247	1146	1308
22	353	722	1027	846	884	1167	1357
941	253	851	1116	945	115	537	1188
221	633	894	1363	196	61	232	834
267	582	1042	1239	352	52	188	632
1279	53	330	875	75	34	145	1352
976	406	982	1059	1098	315	775	972
26	94	176	316	673	412	1142	1144
751	462	925	967	1259	279	554	1087
72	973	1095	1276	459	173	276	1141
1070	486	549	840	1312	277	822	1226
157	425	776	856	216	174	614	1076
769	138	912	1253	1356	833	981	1307
1173	678	1099	1336	653	570	576	959
783	460	657	1050	1108	558	809	1153

193	218	264	349	201	735	920	1314
1047	555	675	956	1085	733	785	921
164	444	1177	1229	92	262	929	1315
1065	280	1013	1201	1187	144	581	825
871	433	743	757	297	370	1158	1260
410	202	245	954	1249	301	597	1347
1292	788	890	995	569	137	485	924
949	545	1035	1088	965	607	731	736
334	616	852	1317	703	659	738	1348
1008	204	414	1154	1346	824	919	923
468	10	1117	1268	516	67	302	496
850	187	314	347	1132	527	559	1213
938	19	843	942	830	422	1073	1250
917	346	552	918	1137	489	988	1109
1326	254	320	991	401	120	492	810
863	598	688	1112	1152	553	1140	1306,
798	529	583	1255	9	842	990	1316
86	139	416	961	392	122	303	464
619	268	699	1208	337	506	517	867
855	93	1094	1335	1320	212	770	1021
681	437	1043	1221	1102	479	626	913
1176	263	742	889	1339	68	150	290
998	387	1240	1296	791	378	497	904
893	329	930	1026	1295	1207	1220	1254
357	146	561	771	66	737	1072	1111
587	599	753	812	882	487	655	1156
154	213	766	1353	251	550	595	807
261	730	732	953	719	534	1113	1164
642	76	1022	1038	113	574	952	1071
696	209	864	1091	911	548	575	966
645	35	662	1321	79	44	689	706
1025	741	841	1334	1041	114	608	1351
935	827	914	1134	313	551	922	1139
327	403	969	1349	441	578	926	1010
510	239	480	603	43	208	752	1163
376	104	704	1264	980	275	808	1086
667	1053	1103	1195	51	252	786	971
385	106	660	1193	903	289	463	505
1286	610	627	1330	143	484	596	734
1192	103	286	968	1218	287	476	739
1283	140	1189	1215	1019	206	571	572
1212	203	488	491	1237	494	927	1251
307	87	108	538	60	246	405	883
1206	478	960	1020	1361	423	442	1266
713	98	116	417	230	525	1011	1074
504	102	207	577	1200	544	955	1157
1289	566	946	962	613	411	656	821
630	185	579	831	1333	573	654	806,

1004	498	835	1015	765	560	661	1037
1245	281	372	690	1275	175	424	677
179	62	379	747	29	707	1128	1202
97	107	565	1214	319	18	615	1267
323	197	590	905	257	45	815	1066
994	217	443	756	1329	602	1133	1194
760	81	233	792	436	386	528	698
746	80	589	1014	874	721	1077	1362
57	304	1203	1358	272	600	1160	1261
729	547	1110	1350	839	461	958	1252
126	393	708	1168	33	536	1051	1238
888	328	989	1093	1127	371	588	814
513	465	847	1129	242	298	367	754
483	274	970	1138	606	244	784	928
1124	30	123	885	365	366	813	1159
543	404	490	820	687	421	495	658
456	69	1067	1309	344	299	556	1114
594	740	951	1155	1225	1049	1075	1143
131	46	1147	1340	17	676	720	755
625	211	415	1219	1058	231	1145	1166
521	291	816	1302	285	288	1009	1162
986	368	1048	1165	1034	369	674	1012
716	151	248	258	101	205	477	805
119	413	526	987	420	243	535	957
1031	282	518	983	172	278	557	832
183	865	1089	1262	533	705	811	1090
40	373	507	977	136	300	580	1313
475	105	402	1268	804	210	950	1161
294	407	868	1246	524	184	493	1265
819	273	546	1092	1305	186	345	823
360	338	691	1060	149	121	377	866
564	601	697	1036	774	631	1115	1227.

m+1 = 3, s = 3, r = 1:

0	1	12	20	26	30	33	35	57,
0	2	40	66	70	24	41	60	52,
0	27	63	3	5	43	69	55	44,
0	34	72	11	56	19	25	32	29,
0	13	15	53	10	37	65	54	6,
0	31	7	4	9	67	48	47	59,
0	21	61	8	18	23	14	45	62,
0	36	42	49	17	51	16	28	46,
0	58	64	22	71	68	50	39	38.

m+1 = 4, s = 3, r = 2:

0	1	37	84	89	328	404	490	558,
0	2	395	178	74	71	168	223	531,
0	491	227	10	55	363	419	488	417,
0	38	81	421	115	463	147	129	173,
0	85	428	301	12	388	23	126	309,
0	559	493	248	450	541	412	527	290,
0	90	229	233	434	565	371	377	121,
0	329	298	188	579	57	437	441	208,
0	405	286	562	278	365	62	103	574.

0 1 37 84 89 328 404 490 558

2	395	178	74	71	168	223	531
491	227	10	55	363	419	488	417
38	81	421	115	463	147	129	173
85	428	301	12	388	23	126	309
559	493	248	450	541	412	527	290
90	229	233	434	565	371	377	121
329	298	188	579	57	437	441	208
405	286	562	278	365	62	103	574,
3	107	324	516	514	152	97	460
532	373	507	118	322	273	133	472
396	536	210	51	150	381	185	263
179	499	423	95	132	96	184	68
75	268	198	318	105	385	424	483
72	69	176	166	221	393	583	529
169	466	283	157	458	242	180	545
224	343	41	216	512	523	500	303,
492	226	187	70	285	387	120	462
418	411	61	80	440	564	427	394
228	247	578	22	146	561	167	9
11	376	73	297	114	289	416	573
56	222	128	232	277	449	125	54
364	433	540	172	530	308	436	362
420	487	300	102	207	526	177	370,
489	557	327	584	88	36	403	83,
39	200	275	518	305	398	98	468
174	254	153	16	235	568	29	7
82	432	448	21	461	415	101	439
422	117	215	392	515	156	50	317
116	64	111	517	355	431	27	28
464	313	554	112	444	250	108	256
148	205	446	477	356	336	142	4
130	195	455	325	390	260	520	65,
86	270	509	154	218	265	181	182
310	123	349	30	408	193	243	510
429	479	359	546	266	161	59	236
302	183	544	471	175	482	262	459
13	238	552	569	576	137	158	219
389	354	414	467	155	204	15	255
24	271	252	33	17	46	191	170
127	369	284	87	296	307	8	410,

560	525	53	186	171	426	375	326
291	549	52	47	548	453	521	367
494	44	571	456	334	292	537	18
249	397	253	110	141	391	20	454
451	400	196	351	550	211	25	78
542	66	504	48	382	34	92	340
413	261	358	368	264	192	136	190
528	380	522	151	272	384	241	131,
91	383	425	547	77	135	109	43
122	295	160	251	203	543	269	237
230	76	162	258	346	257	294	341
234	430	259	100	445	312	49	199
435	246	35	386	60	231	486	113
566	505	484	360	347	331	314	338
372	67	106	267	342	535	465	165
378	163	213	555	280	93	319	480,
330	99	345	212	134	159	485	534
209	40	282	197	323	220	506	498
299	276	288	79	119	577	361	402
189	570	339	240	399	140	452	374
580	315	497	469	496	401	533	239
58	551	508	14	348	470	306	32
438	26	274	553	567	316	519	476
442	144	352	473	138	332	581	201,
406	502	538	501	244	320	474	5
575	407	45	217	409	481	353	478
287	143	475	495	344	31	281	139
563	124	556	145	206	539	572	225
279	337	245	293	42	311	164	202
366	357	503	19	524	350	379	333
63	443	447	6	335	304	194	214
104	149	321	457	513	511	94	582.

$m+1 = 3,\ s = 4,\ r = 1$:

0 1 18 46 55 69 131 151 170
175 181 183 210 217 248 258 270,

0 2 89 267 92 138 93 77 161
29 67 36 223 262 147 110 243,

0 211 79 188 160 86 139 127 143
52 20 117 142 50 197 39 44,

0 259 106 148 204 189 114 141 250
205 179 62 82 201 101 112 222,

0 271 34 65 27 159 75 91 136
90 265 87 241 108 145 260 221,

0 70 185 16 61 196 12 232 146
225 263 15 198 166 33 84 190,

0 176 94 156 80 200 25 43 206
22 71 195 26 235 208 242 10,

0 152 192 230 113 252 199 133 157
255 240 28 165 37 163 51 256,

0 19 119 177 122 32 66 24 30
97 140 253 123 168 107 59 191,

0 132 38 60 7 63 214 233 64
81 48 238 109 246 244 194 118,

0 56 31 74 102 187 226 239 207
41 57 266 125 111 237 53 231,

```
0  182   17  247  209  150   54  272  130
    68  216   45  257  169  269  180  174,

0   47   13  234   11  104  103   88  172
    78  121   40  158  254  100  149    5,

0  184  220   21  134  178  154  251    4
   186   49   72  173   58  261  213    3,

0  218  193  120   96  115  155  264  215
   236   14  126   76  203  162  219  128,

0  171  227  137  228  224  129  212  164
   124    9  202  135   85  245   23  105,

0  249    8   95    6   42   35  229  268
   116  144  153   98   99   83   73  167.
```

REFERENCES

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

2. Blahut, R.E., Theory and Practice of Error Control Codes, Addison-Wesley, Reading, MA., 1983.

3. Blake, I.P. and R.C. Mullin, An Introduction to Algebraic and Combinatorial Coding Theory, Academic Press, New York, 1976.

4. Blythe, J.H. and K. Edgecombe, "Net Coding Gain of Error-Correcting Codes," Proc. IEE, vol. 122, No. 6, pp. 609-614, June 1975.

5. Chen, C.L., "On Majority-Logic Decoding of Finite Geometry Codes," IEEE Trans. Inform. Theory, vol. IT-17, No. 3, pp. 332-336, May 1971, and vol. IT-18, No. 4, pp. 539-541, July 1972.

6. Cheung, C.S.L., Characteristic M-Sequence Codes Over GF(q), M.A.Sc. Thesis, University of Ottawa, Ottawa, Ontario, 1983.

7. Clark, G.C., Jr. and J.B. Cain, Error-Correction Coding for Digital Communications, Plenum Press, New York, 1981.

8. Delsarte, P., "A Geometrical Approach to a Class of Cyclic Codes," J. Combinatorial Theory, 6, pp. 340-358, 1969.

9. Delsarte, P., J.M. Goethals and F.J. MacWilliams, "On Generalized Reed-Muller Codes and Their Relatives," Infor. Control, 16, pp. 403-442, July 1970.

10. Goethals, J.M. and P. Delsarte, "On a Class of Majority-Logic Decodable Cyclic Codes," IEEE Trans. Inform. Theory, vol. IT-14, No. 2, pp. 182-188, March 1968.

11. Goethals, J.M., "Some Combinatorial Aspects of Coding Theory," in A Survey of Combinatorial Theory, ed. by J.N. Srivastava, North-Holland, Amsterdam, 1973.

12. Goethals, J.M., "Threshold Decoding -- A Tentative Survey," in Coding and Complexity, G. Longo ed., Springer-Verlag, New York, 1975.

13. Gorenstein, D.C. and N. Zierler, "A Class of Error-Correcting Codes in p^m Symbols," J. Soc. Indus. Appl. Math., 9, pp. 207-214, June 1961.

14. Graham, R.L. and J. MacWilliams, "On the Number of Information Symbols in Difference-Set Cyclic Codes," Bell Syst. Tech. J., vol. 45, pp. 1057-1070, September 1966.

15. Hamada, N., "The Rank of the Incidence Matrix of Points and d-flats in Finite Geometries," J. Sci. Hiroshima Univ., vol. 32, pp. 381-396, 1968.

16. Hamming, R.W., "Error Detecting and Error Correcting Codes," Bell Syst. Tech. J., vol. 29, pp. 147-160, April 1950.

17. Hartmann, C.R.P., J.B. Ducey and L.D. Rudolph, "On the Structure of Generalized Finite Geometry Codes," IEEE Trans. Inform. Theory, vol. IT-20, No. 2, pp. 240-252, March 1974.

18. Huang, J.F., S.G.S. Shiva and G. Seguin, "On Certain Projective Geometry Codes," IEEE Trans. Inform. Theory, vol. IT-30, No. 2, pp. 385-388, March 1984.

19. Kasami, T., S. Lin and W.W. Peterson, "New Generalizations of the Reed-Muller Codes, Part I: Primitive Codes," IEEE Trans. Inform. Theory, vol. IT-14, No. 2, pp. 189-199, March 1968.

20. Kasami, T. ad S. Lin, "On Majority-Logic Decoding for Duals of Primitive Polynomial Codes," IEEE Trans. Inform. Theory, vol. IT-17, No. 3, pp. 322-331, May 1971.

21. Lidl, R. and H. Niederreiter, Finite Fields, Addison-Wesley, Reading, MA., 1983.

22. Lin, S., "On the Number of Information Symbols in Polynomial Codes," IEEE Trans. Inform. Theory, vol. IT-18, No. 6, pp. 785-794, November 1972.

23. Lin, S. and D.J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliff, NJ., 1983.

24. Lucky, R.W., J. Salz and E.J. Weldon, Jr., Principles of Data Communications, McGraw-Hill, New York, 1968.

25. MacWilliams, F.J. and H.B. Mann, "On the p-Rank of the Design Matrix of a Difference Set," Infor. Control, 12, pp. 474-488, 1968.

26. MacWilliams, F.J. and N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland, Amsterdam, 1977.

27. Massey, J.L., Threshold Decoding, MIT Press, Cambridge, MA., 1963.

28. McEliece, R.J., The Theory of Information and Coding, Addison-Wesley, Reading, MA., 1979.

29. Muller, D.E., "Application of Boolean Algebra to Switching Circuit Design and to Error-Correction," IRE Trans. Electronic Computers, vol. EC-3, pp. 6-12, 1954.

30. Muscati, Z.A. and S.G.S. Shiva, "1-step Majority-Logic Decoding with Two Rounds of Estimation," Proc. ICC '81, Denver, pp. 65.7.1-65.7.5, 1981.

31. Peterson, W.W. and E.J. Weldon, Jr., Error-Correcting Codes, 2nd Edition, MIT Press, Cambridge, MA., 1972.

32. Pless, V., Introduction to the Theory of Error-Correcting Codes, Wiley, New York, 1982.

33. Prange, E., "Some Cyclic Error-Correcting Codes With Simple Decoding Algorithms," AFCRC-TN-58-156, Air Force Cambridge Research Center, Bedford, Mass., April 1958.

34. Rao, C.R., "Cyclical Generation of Linear Subspaces in Finite Geometries," in Combinatorial Mathematics and Its Applications, R.C. Bose and T.A. Dowling eds., Univ. of North Carolina Press, Chapel Hill, NC., 1969.

35. Reed. I.S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," IEEE Trans. Inform. Theory, vol. IT-4, pp. 38-49, 1954.

36. Rudolph, L.D., Geometric Configuration and Majority-Logic Decodable Codes, M.E.E. Thesis, University of Oklahoma, Norman, Okla., 1964.

37. Rudolph, L.D., "A Class of Majority-Logic Decodable Codes," IEEE Trans. Inform. Theory, vol. IT-13, pp. 305-307, April 1967.

38. Rudolph, L.D. and C.R.P. Hartmann, "Decoding by Sequential Code Reduction," IEEE Trans. Inform. Theory, vol. IT-19, No. 4, pp. 549-555, July 1973.

39. Rudolph, L.D., "Some Current Research in Decoding Theory," in Coding and Complexity, G. Longo ed., Springer-Verlag, New York, 1975.

40. Shannon, C.E., "A Mathematical Theory of Communication," Bell Syst. Tech. J., vol. 27, pp. 379-423 and pp. 623-656, July and October 1948.

41. Shiva, S.G.S. and S. E. Tavares, "On Binary Majority-Logic Decodable Codes," IEEE Trans. Inform. Theory, vol. IT-20, No. 1, pp. 131-133, January 1974.

42. Slepian, D., "A Class of Binary Signalling Alphabets," Bell Syst. Tech. J., vol. 35, pp. 203-234, January 1956.

43. Slepian, D., "Some Further Theory of Group Codes," Bell Syst. Tech. J., vol. 39, pp. 1219-1252, September 1960.

44. Smith, K.J.C., "On the p-Rank of the Incidence Matrix of Points and Hyperplanes in a Finite Projective Geometry," J. Combinatorial Theory, 7, pp. 122-129, 1969.

45. van Lint, J.H., Introduction to Coding Theory, Springer-Verlag, New York, 1982.

46. Weldon, E.J., Jr., "Difference-Set Cyclic Codes," Bell Syst. Tech. J., vol. 45, pp. 1045-1055, September 1966.

47. Weldon, E.J., Jr., "New Generalizations of the Reed-Muller Codes, Part II: Non-Primitive Codes," IEEE Trans. Inform. Theory, vol. IT-14, No. 2, pp. 199-205, March 1968

48. Weldon, E.J., Jr., "Some Results on Majority-Logic Decoding," in Error Correcting Codes, H.B. Mann ed., Wiley, New York, 1969.