CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES 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 a poor 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.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

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 mauvaise qualité.

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.

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

NL-339 (r. 82/08)

Canada

CHARACTERISTIC M-SEQUENCES

by

CHEUNG, Cecillia Siu-Ling

A thesis
presented to University of Ottawa
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the
Department of Electrical Engineering

OTTAWA, Ontario, 1983

University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

The characteristic m-sequences over GF(q) with $q=2^s$ are discussed in this thesis. Relevant finite field theory is first quoted, then general aspects of cyclic codes and m-sequences are briefly introduced. The factoring of binary primitive irreducible polynomials over GF(q) is discussed and illustrated by some examples. By using one of these q-nary primitive irreducible factors, characteristic m-sequences over GF(q) can be generated methodically. This is illustrated with some examples. Theorems are derived with respect to the special case of $GF(q^2)$. Properties of these q-nary characteristic m-sequences are then observed and finally discussions on their unique error-correcting capability and majority logic decodable capability are included.

ACKNOWLEDGEMENTS

The author wishes to express her deepest gratitude to Professor Saligram G.S. Shiva for his guidance and encouragement throughout this research. Thanks are due to Mr. Jen-Fa Huang for his helpful discussions and suggestions, and for his assistance with the computer programming. Thanks are also due to Mr. Ted Anderson for his many helpful suggestions. The financial support of the National Sciences and Engineering Research Council of Canada is gratefully acknowledged.

CONTENTS

Chapter I

INTRODUCTION

1.1 HISTORICAL DESCRIPTION OF CYCLIC CODES AND M-SEQUENCES

In the mid-1940's, N. Wiener, C.E. Shannon and R.M. Fano were among the pioneers who initiated the science of statistical communication theory. Error-correcting codes were first examined by R. Hamming[9], with major specific contributions by I.S. Reed, D.E. Muller, M. Golay, and others. They were finally systematized by D. Slepian [18] in 1954. Cyclic codes were first studied by Prange[16] in 1957. The discovery of BCH codes by Hocquenghem in 1959 and independently by Bose and Chaudhuri in 1960, had important subsequent contributions on random error-correcting codes. The BCH codes are cyclic codes. They were first defined in binary symbols and then were generalized to codes in p^m symbols (where p is any prime and m is any positive integer) by Gorenstein and Zierler [21] in 1960. Cyclic codes are attractive for two reasons: first, encoding and syndrome calculation of a cyclic code can be implemented easily by employing simple shift registers with feedback connections; secondly, because of their considerable inherent algebraic structure, it is possible to find simple and efficient decoding methods.

Maximum-length-sequence codes. (often called m-sequences and will be hereafter) comprise one important subset of cyclic codes. Binary m-sequences were well investigated in the 1960's and some results on their structural properties, methods of generation, correlation properties, decoding procedures, and applications to various electronic systems were obtained. The majority logic decoding is an effective decoding scheme for certain classes of cyclic codes such as m-sequences. The first majority logic decoding algorithm was devised in 1954 by I.S. Reed[17] for a class of multiple error-correcting codes discovered by D.E. Muller. Reed's algorithm was later extended and generalized by many coding investigators. The first unified formulation of majority logic decoding algorithms was due to J.L. Massey[12] in 1963. It is well known that all binary m-sequences are one-step majority logic decodable.

1.2 APPLICATIONS OF M-SEQUENCES AND A BRIEF SUMMARY OF THE THESIS

Binary m-sequences have been used extensively in many communication systems [5][7][19] because of their unique algebraic structure and simple method of generation using shift register. By definition, binary m-sequences are the longest sequences that can be generated by a given shift register or a delay element of a given length. Binary m-sequences are used in communications and ranging as well as

spread spectrum systems. Other codes can do no better than equal their performance. Therefore it is proper that they and some of their applications be given adequate exposure.

Their first application is as a source of random numbers. Certain computer simulations and other applications require a program to behave probabilistically, and the probabilistic behavior usually is produced by selecting a random number. The successive digits produced by some linear maximal feedback shift registers, the m-sequences, satisfy the specifications for the random numbers, and therefore they have been used in random number generator subroutines. In many communication systems where random numbers are required, m-sequences are used. They are also used in applications of direct sequences and frequency hopping in spread spectrum systems.

The second application of long m-sequences is for security in data transmission systems where the data is of a sensitive nature. A message must be sent from the sender to receiver, but it is possible an eavesdropper may listen to all messages during transmissions. It is the purpose to find a simple cryptographic system to encode messages between sender and receiver, so that the eavesdropper will not understand what he has heard. Yet it is required that the encoding and decoding operations of the message be relatively simple. Indeed the cryptographic systems have

been stimulated by diplomatic and military needs. Of late, data communications of a non-military nature have grown substantially, and with it has grown the need for protecting data from unauthorized access. Figure 1.1 shows a crude block diagram of a secure communication system where m-sequences are used.

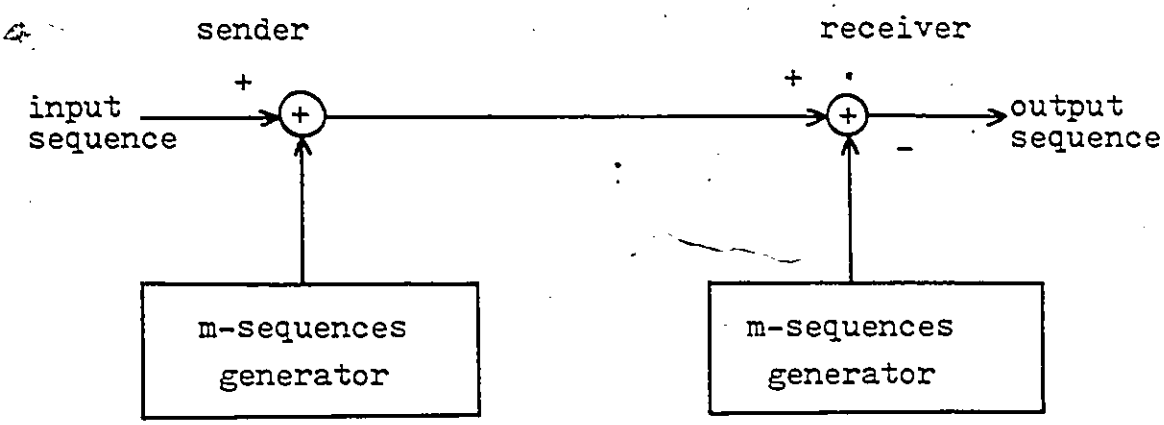


Fig. 1.1: A secure communication system based on an m-sequence.

In such systems, the sender adds an m-sequence to the data before transmission, and the receiver subtracts the same m-sequence from the received sequence. The decoded sequence then is identical to the original transmitted data, but an uninformed observer receives the sum sequence which appears to be quite random because of the nature of m-sequences. Without knowing the m-sequence used the observer cannot decode the data. Thus the data transmission system is secure from casual eavesdroppers.

The third application of m-sequences is to radar ranging systems. A pulse train (or continuous wave) transmitted from the radar to a distant body such as the moon or a planet, and the signal reflected from the body returns to the receiver on earth sometime later. Since the pulse train travels at a known speed (the speed of light), one can compute the distance of the body from the earth by measuring the time delay between pulse transmission and pulse reception. In environments with extreme background noise, a shift register modulated pulse train, using an m-sequence, has the property that its autocorrelation function is recoverable despite a noise-to-signal excess of many decibels. In practice, for example, the Venus ranging experiment of M.I.T.'s Lincoln Laboratory in 1959 and 1961 employed an m-sequence of length $2^{13}-1=8191$ to determine whether to transmit 'pulse' or 'no pulse' in consecutive time intervals.

There has been much research done on binary m-sequences and as mentioned above, they are widely used in many communication systems. However, for a sophisticated communication system, one would like to reduce the transmission bandwidth of a sequence, or a signal, by reducing the number of bits required to transmit such a sequence. Thus one can accomodate more signals at once. Consider the q-nary m-sequences, where there are $q=2^s$ levels instead of two levels as in the binary case. One can

transmit q-nary m-sequences with length per unit time s times larger than transmitting binary m-sequences. Consequently, the bandwidth required to transmit q-nary m-sequences is compressed by a factor of s. It is useful when bandwidth limitation is taking into account and also when one wants to transmit more signals through the same channel. All binary m-seuences are one-step majority logic decodable, it is important because majority logic decoding scheme has been well established. However, it is found that not all q-nary m-sequences are one-step majority decodable but all q-nary characteristic m-sequences [20] (will be abbreviated c-m-sequences hereafter) are. Therefore, studies of q-nary c-m-sequences are given in this thesis. Furthermore, when information is in the form of non-binary basis, or when errors occur in bursts, it is more efficient to use q-nary codes.

Basic finite field theory and concepts of minimal polynomials are quoted as a basis for this thesis in Chapter II. General aspects of cyclic codes and m-sequences are introduced and properties of binary m-sequences are discussed. Added also are definitions of weight, distance and error-correcting capability of a code sequence. Chapter III, IV and V are the main concern of the thesis. Chapter III gives the method of factoring binary primitive irreducible polynomials over GF(q), where $q=2^s$. General approach is first introduced and followed by some

illustrations. Tabulated results are listed for factorizations of different degrees of binary primitive irreducible polynomials and/or of different Galois fields. A theorem is derived with respect to the $GF(q^2)$ case and proof of this theorem is provided. Chapter IV deals with the construction of c-m-sequences over GF(q). By using one primitive irreducible factor over GF(q) obtained from Chapter III as the logic provided for GF(q), c-m-sequences can be constructed methodically over GF(q) by using simple linear feedback shift registers. A table is provided for constructing various lengths of c-m-sequences over different Galois fields. Another theorem is derived with respect to the $GF(q^2)$ case. In Chapter V, general aspects of q-nary c-m-sequences are observed and discussed. The properties of such sequences are discussed first, they are indeed the generalized properties of binary m-sequences, then their importance on error-correcting and majority logic decodable capabilities. Finally, conclusions upon these c-m-sequences over GF(q) are given in Chapter VI.

Chapter II

REVIEW OF CYCLIC CODES AND M-SEQUENCES

There are many good references on cyclic codes and m-sequences, such as Peterson & Weldon[14], S. Lin[10], Berlekamp[2], Clark[4], Pless[15], Gallager[6], MacWilliams & Sloane[11], McEliece[13], and Golomb[8]. By using these as a basis, general aspects of cyclic codes and m-sequences are given in this chapter.

2.1 GALOIS FIELD AND MINIMAL POLYNOMIAL

Since algebraic structure[1] has been the basis of the most known codes such as cyclic codes as well as m-sequences, it is essential to introduce first the finite field theory.

A field is a set of elements closed under two the operations of 'addition' (denoted by +) and 'multiplication' (denoted by *), and containing additive and multiplicative identity element denoted by '0' and '1' respectively. The addition and multiplication are associative and commutative. For arbitrary field elements a, b and c, the distributive law of multiplication over addition holds, i.e., a * (b + c)

= a * b + a * c. Every field element a has a unique additive inverse denoted -a such that a+(-a)=0. Every nonzero element a has a unique multiplicative inverse denoted 1/a such that a*(1/a)=1.

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 GF(q). For example, when q=2, GF(2)={0,1}. The field is called the Galois field of two elements or simply a binary field. Finite fields exist only when the number of elements is a prime number or a power of a prime number, the former are called prime or gound fields, while the latter are called extension fields over the prime fields. Any two finite fields of the same order are isomorphic, i.e., Galois fields of a given order are essentially unique.

In coding theory, a code word or a code vector is usually expressed in polynomial form. A polynomial

$$f(x) = f_n x^n + f_{n-1} x^{n-1} + \ldots + f_1 x + f_0 \qquad (2.1.1)$$

over GF(q) with degree n, $f_n \neq 0$, is said to be monic if $f_n=1$. The reciprocal polynomial of f(x), denoted by $f^*(x)$, is defined as

$$f^*(x) = x^n f(x^{-1}) . \qquad (2.1.2)$$

A polynomial f(x) is said to be irreducible over GF(q) if f(x) cannot be factored using only elements from GF(q). The same polynomial, however, can always have roots in an extension field. Any monic polynomial can be uniquely factored into monic irreducible polynomials over GF(q). A polynomial f(x) over GF(q), of degree n, is said to have exponent e if it divides $1-x^e$ and not any other $1-x^{e'}$ for any $e'<e$; and f(x) has at most n roots in any field containing GF(q).

If a field contains an element α, the least positive integer r for which $\alpha^r=1$ is called the order of α. The order of an element α is also the number of distinct powers of this element. (The order of 0 is undefined or sometimes referred to as $-\infty$.) For example, let q=8, and α be an element of GF(q), $\alpha \neq 0$ or 1, then

$$GF(8) = \{0,1,\alpha,\alpha^2,\alpha^3,\alpha^4,\alpha^5,\alpha^6\} \qquad (2.1.3)$$

and $\alpha^7=1$. If α has an order r then $\alpha^{r'}=1$ iff r' is a multiple of r. In a field of order q, α is a primitive field element iff the order of α is q-1. A finite field of order q must contain a primitive field element whose order is q-1 and whose powers include all nonzero field elements.

In general, if α is a primitive element of $GF(p^m)$ where p is a prime and m is any positive integer, an irreducible

polynomial p(x) of degree m which has α as a root is called a primitive polynomial.

Consider a prime field GF(p) and an extension field $GF(p^m)$, and let α be any element of the extension field. The monic polynomial m(x) of smallest degree with coefficients in GF(p) such that $m(\alpha)=0$ is called the minimal polynomial or minimum function of α. The minimum function m(x) of any element is irreducible over GF(p). α, α^p have the same minimal polynomial. Every element of an extension field of degree m over GF(p) has a minimum function of degree m or less. The minimal polynomial of a primitive element of $GF(p^m)$ has degree m.

It is necessary to introduce the concept of cyclotomic cosets now. If α is an element of $GF(p^m)$, α and α^p have the same minimal polynomial in GF(p). It leads to the fact that the powers of α fall into disjoint sets which are called cyclotomic cosets.

A cyclotomic coset C_s containing s as the smallest element in this set, over $GF(p^m)$, is defined as

$$C_s = \{s, p_1, p_2, \cdots, p_{j-1}\} \quad , \tag{2.1.4}$$

where $p_i = (p_{i-1} \, s) \bmod n,$

$$n = p^m - 1 \, , \tag{2.1.5}$$

and j is the smallest positive integer such that

$$(p_j\ s) \bmod n = s\ . \qquad (2.1.6)$$

Therefore j=m-1 and $p_j=p_{m-1}=s$. (Recall that a modulo b = residue after dividing a by b.)

The subscript s, which is the smallest element in the coset, is called the coset leader.

For example, let p=2 and m=4, the cyclotomic cosets modulo n=15 are as the following:

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1, 2, 4, 8\} \\ C_3 &= \{3, 6, 12, 9\} \qquad (2.1.7) \\ C_5 &= \{5, 10\} \\ C_7 &= \{7, 14, 13, 11\}\ . \end{aligned}$$

Each coset leader s has a corresponding minimal polynomial $m_s(x)$ for which $\alpha^s \in GF(p^m)$. There are various methods to calculate $m_s(x)$ such as the method given by Berlekamp[2]. However, there are also tables of binary irreducible polynomials available. Peterson & Weldon [14, Appendix C] have an extensive table of binary irreducible polynomials of degrees up to 34. For example, when p=2 and m=4, corresponding to GF(2^4), information obtained from the table is,

DEGREE 4 1 23F 3 37D 5 07 .

The first entry '1' corresponds to the coset leader '1', it means that α is a root of the polynomial, by converting the following entry '23' to binary digits 010 011, the polynomial can be written as $m_1(x)=1+x+x^4$. The letter 'F' which follows '23' means that the corresponding minimal polynomial $m_1(x)$ is a primitive polynomial; since α is a root of $m_1(x)$, $1+\alpha+\alpha^4=0$, this can be used as the logic to generate GF(2^4). A detailed description of the meanings of different letters is given in Peterson & Weldon [14]. Consider now the next entry '3', which means that α^3 is a root; then the entry '37' can be converted into binary form as 011 111, therefore the corresponding minimal polynomial is $m_3(x)=1+x+x^2+x^3+x^4$. The letter 'D' means that the roots of $m_3(x)$ are linearly independent. The next entry '5' means that α^5 is a root; '07' in binary form is 000 111, it follows that $m_5(x)=1+x+x^2$. By examining (2.1.7), there should be one more minimal polynomial, $m_7(x)$, with α^7 as a root. Since there is no more information given from the table, it can be concluded that the corresponding polynomial $m_7(x)$ must be a reciprocal of one of the polynomials $m_1(x)$, $m_3(x)$ or $m_5(x)$. In order to find out which polynomial is the reciprocal polynomial of $m_7(x)$, α^{-7} and α^{-14} are calculated,

$$\alpha^{-7} = \alpha^{15-7} = \alpha^8$$

$$\alpha^{-14} = \alpha^{15-14} = \alpha \quad .$$

The result implies that $m_7(x)$ is the reciprocal of $m_1(x)$, or $m_7(x)=m_1^*(x)=1+x^3+x^4$. Therefore, Table 2.1 is formed and it gives all minimal polynomials in GF(2^4).

Elements	Corresponding Minimal Polynomials
{0}	$m_{-\infty}(x)=x$
{1}	$m_0(x)=1+x$
$\{\alpha, \alpha^2, \alpha^4, \alpha^8\}$	$m_1(x)=1+x+x^4$
$\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\}$	$m_3(x)=1+x+x^2+x^3+x^4$
$\{\alpha^5, \alpha^{10}\}$	$m_5(x)=1+x+x^2$
$\{\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}\}$	$m_7(x)=1+x^3+x^4=m_1^*(x)$

Table 2.1: Minimal polynomials of elements in GF(2^4) defined by $1+\alpha+\alpha^4=0$.

The exponent e to which the minimum function $m_j(x)$ belongs, is given by

$$e = \frac{2^m - 1}{\text{GCD}\ (2^m-1,\ j)} \quad , \qquad (2.1.8)$$

where GCD(a,b) denotes the greatest common divisor of a and b. If GCD(a,b)=1, a and b are said to be relatively prime. In equation (2.1.8), if $GCD(2^m-1,j)=1$, the corresponding $m_j(x)$ is a primitive polynomial of degree m and $m_j(x)$ has the highest exponent 2^m-1. The number of binary polynomials of degree m which have the highest exponent 2^m-1 is given by,

$$\theta(m) = \frac{\phi(2^m-1)}{m} \qquad (2.1.9)$$

where $\phi(2^m-1)$ is the Euler number of 2^m-1, the number of positive integers that are relatively prime to and less than 2^m-1.

$x^{p^m}-x$ is the product of all monic polynomials irreducible over GF(p) whose degrees divide m. For example, in the case of p=2 and m=4,

$$x^{16}-x = x(1+x)(1+x+x^4)(1+x+x^2+x^3+x^4)$$
$$(1+x+x^2)(1+x^3+x^4) \qquad (2.1.10)$$

This is an important property because factoring x^n-1 where $n=p^m-1$, is important for constructing cyclic codes.

2.2 GENERAL ASPECTS OF CYCLIC CODES AND M-SEQUENCES

An (n,k) cyclic code C over GF(q) is a linear code with the special property that any cyclic shift of a code word is another code word. That is, if an n-tuple

$$V=(v_0,v_1,v_2,\ \cdots\ ,v_{n-1}) \tag{2.2.1}$$

is a code vector of C, the n-tuple obtained by shifting V by 'i' places cyclically,

$$V^{(i)}=(v_{n-i},v_{n-i+1},\cdots,v_{n-1},v_0,\cdots,v_{n-i-1}) \tag{2.2.2}$$

is also a code vector. In polynomial representation,

$$V(X)=v_0+v_1X+v_2X^2+\ldots+v_{n-1}X^{n-1}, \tag{2.2.3}$$

and

$$V^{(i)}(X)=v_{n-i}+v_{n-i+1}X+\ldots+v_{n-1}X^{i-1}+v_0X^i +v_1X^{i+1}+\ldots+v_{n-i-1}X^{n-1}. \tag{2.2.4}$$

An (n,k) cyclic code can be completely specified by one polynomial g(X) of degree n-k and exponent n, which is known as generator polynomial of the cyclic code. Every code polynomial V(X) is a multiple of g(X), and every polynomial of degree n-1 or less which is a multiple of g(X) must be a code polynomial.

The generator polynomial g(X) is a factor of X^n-1, that is,

$$X^n-1=g(X)h(X)\ , \tag{2.2.5}$$

which implies that if g(X) is a polynomial of degree n-k and is a factor of X^n-1, then g(X) generates an (n,k) cyclic code. The polynomial h(X) is known as the parity check or recursion polynomial and has degree k.

By using the procedures mentioned in last section, all factors of X^n-1 can be obtained for various n. For example, when m=4, $n=2^4-1=15$, and from equation (2.1.10), $X^{15}+1$ can be factored as

$$X^{15}+1=(1+X)(1+X+X^4)(1+X+X^2+X^3+X^4)(1+X+X^2)(1+X^3+X^4) . \quad (2.2.6)$$

The generator polynomial must contain at least one of these minimal polynomials which has the highest exponent n, implying that g(X) also has exponent n. If I(X) denotes the k-bit information polynomial,

$$I(X)=i_0+i_1X+i_2X^2+\ldots+i_{k-1}X^{k-1}, \quad (2.2.7)$$

then the encoding of I(X) into the corresponding code polynomial V(X) is given by,

$$V(X)=g(X)I(X) \bmod (X^n+1) . \quad (2.2.8)$$

Cyclic codes are most easily implemented with shift register devices. The cyclic code properties and the property that each code polynomial is a multiple of the generator polynomial, minimizes the storage facilities for the encoding dictionary. A k-stage shift register is a

device consisting of k consecutive memory cells. The content of each cell shifts to the next cell in time to the regular beat of a clock or by means of other timing device. According to the parity check polynomial h(x), which is given by (2.2.5) if the generator polynomial is specified, suitable feedback function can be fed back into the leftmost cell of the shift register assuming the shifting direction is from left to right. Figure 2.1 shows a block diagram of such a shift register device with feedback mechanism, where mc's denote memory cells and the number of mc's increases

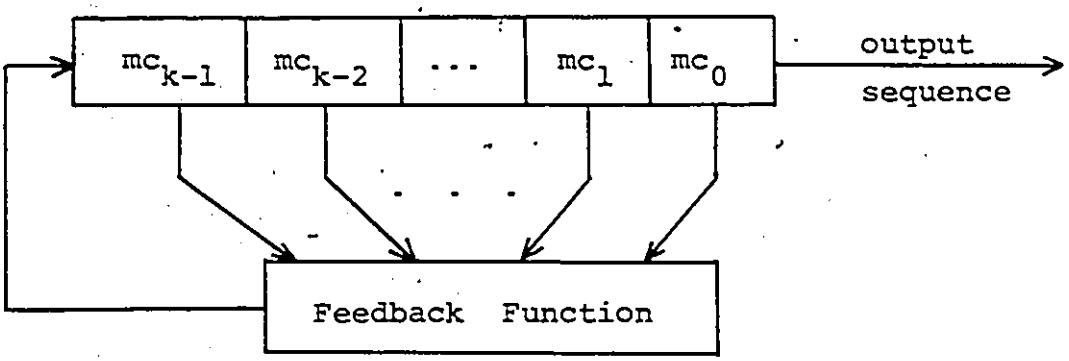


Fig. 2.1: A block diagram of shift register device with feedback connection.

from right to left.

An (n,k) cyclic code is completely specified by its generator polynomial g(X), which is a factor of X^n+1. According to equation (2.2.5), let the polynomial h(X) with degree k be

$$h(X)=h_0+h_1X+h_2X^2+ \ldots +h_kX^k \ , \qquad (2.2.9)$$

where $h_0=1$ and $h_k=1$. Let $V(X)=v_0+v_1X+v_2X^2+\ldots+v_{n-1}X^{n-1}$ be a code polynomial, it can be shown [10] that

$$v_{n-k-j} = \sum_{i=0}^{k-1} h_i \, v_{n-i-j} \quad \text{for } 1 \le j \le n-k \ . \qquad (2.2.10)$$

Given the k information digits, equation (2.2.10) is a rule to determine the n-k parity check digits of the code polynomial V(X). Thus, the (n,k) cyclic code generated by g(X) is also completely specified by $h(X)=(X^n+1)/g(X)$.

The encoding procedures of an (n,k) cyclic code by using a shift register device with feedback function defined by the parity check polynomial h(X) are illustrated by an example. The (7,4) Hamming code is chosen. It is easy to obtain all factors of X^7+1 by same approach which was discussed in last section,

$$X^7+1=(X+1)(X^3+X+1)(X^3+X^2+1) \ . \qquad (2.2.11)$$

It is desired to have $g(X)=X^3+X+1$, thus g(X) has exponent 7. From (2.2.5), h(x) can be determined as

$$h(X)=(X^7+1)/(X^3+X+1) =X^4+X^2+X+1. \ . \qquad (2.2.12)$$

The shift register device shown in Figure 2.2 can be used for encoding the (7,4) Hamming code defined by (2.2.12). At first, the initial infomation bits (a 4-tuple) are loaded on

the shift register, and after four shifts the information bits are stored in the register. Since the inputs to the binary adder correspond to the nonzero terms of h(X), its output is the first parity check bit. On the fifth shift this digit is entered in the leftmost stage of the register and the first information bit is shifted out. Because the code is cyclic, the second parity check digit is the sum of the digits in the same positions relative to itself as the first parity check digit. The third parity check digit is calculated in the similar manner. In order to illustrate these operations, let an initial information 4-tuple (1,1,0,1) be loaded on the shift register, and the corresponding clock beat as '1'. Table 2.2 shows the contents of the shift register corresponding to each clock beat.

The output sequence, which corresponds to the rightmost digit of each clock beat, is found to be 1101001, with the sequence repeating itself after the seventh shift. This is a practical method to generate a binary linear (n,k) cyclic code. Therefore with an initial k-tuple $(i_0, i_1, \ldots, i_{k-1})$ loaded on the k-stage shift register and with a feedback function defined by the parity check polynomial h(X), an n-tuple code word $(v_0, v_1, \ldots, v_{n-1})$ can be generated. The first k bits are the information bits which remain unaltered while the last n-k bits are the parity check bits.

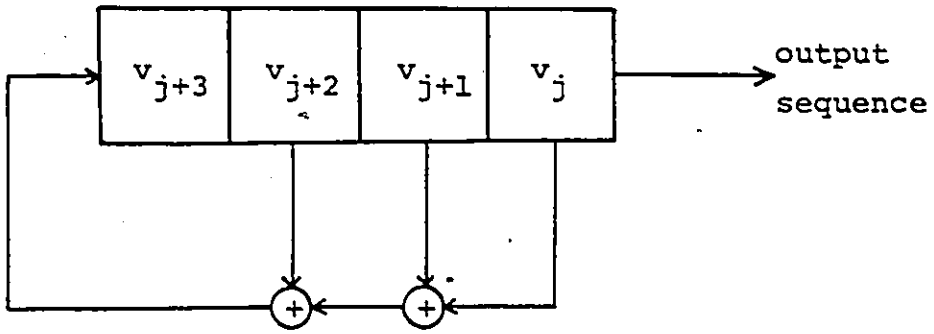


Fig. 2.2: A 4-stage shift register for encoding the (7,4) code defined by $h(X)=X^4+X^2+X+1$.

Time	$(v_{j+3}, v_{j+2}, v_{j+1}, v_j)$
1	1 0 1 1
2	0 1 0 1
3	0 0 1 0
4	1 0 0 1
5	1 1 0 0
6	1 1 1 0
7	0 1 1 1
8	1 0 1 1
9	0 1 0 1
	(repeating)

Table 2.2: Contents of shift register of Figure 2.2 with initial condition (1,0,1,1).

From equation (2.2.8), a code word can be generated by the corresponding generator polynomial g(X). However, from (2.2.10) and the shift register scheme mentioned above, a code word can also be constructed by the parity check polynomial h(X). In order to avoid ambiguity, in this thesis, if a code word V(X) is said to be generated by g(X), equation (2.2.8) is applied; if V(X) is said to be generated (or constructed) by h(X), it means that the feedback connection of a shift register device, which is used to construct V(X), is determined by the parity check polynomial h(X).

In general, if a code word $V(X)=v_0+v_1X+v_2X^2+\ldots+v_{n-1}X^{n-1}$ is generated by the generator polynomial g(X), then the reciprocal polynomial of V(X), $V^*(X)=v_{n-1}+v_{n-2}X+\ldots+v_0X^{n-1}$ is generated by the reciprocal polynomial of g(X), g*(X). It is true also for the case of the parity check polynomial; if V(X) is generated by h(X), then V*(X) is generated by h*(X), the reciprocal polynomial of h(X).

The (Hamming) weight of an n-tuple V, w(V), is defined as the number of nonzero components of V. The (Hamming) distance between two n-tuples U and V, d(U,V), is defined as the number of components in which they differ. It is obvious that under the modulo-2 addition for binary linear

codes, $d(U,V)=w(U,V)$. Given a linear code, the distance between all possible pairs can be calculated. The smallest distance is called the minimum distance of that code, which is denoted as d_{min}. In general, a code with minimum distance d_{min} has error-correcting capability

$$t=[(d_{min}-1)/2] , \qquad (2.2.13)$$

where $[(d_{min}-1)/2]$ denotes the largest integer no greater than $(d_{min}-1)/2$. Minimum distance of a code is a very important factor because it determines the error-correcting capability of that code.

Consider now an important subset of the cyclic codes, known as the maximum-length-sequence codes, or simply the m-sequences. They have a number of uses in many communication systems. M-sequences are, by definition, the longest codes that can be generated by a given shift register or a delay element of a given length. An m-sequence of length q^k-1 is a sequence of elements of GF(q) that appears in a given stage of a k-stage primitive shift register over GF(q) as it cycles through the nonzero states on its maximum-length cycle. In order to construct an m-sequence, a primitive polynomial h(x) is used as the parity check polynomial, and it determines the feedback connection of the k-stage shift register. The maximum period possible for a linear feedback shift register of k-stages is q^k-1. The sequence will start repeating as soon as the vector stored in the shift register

repeats. Therefore there must be a distinct vector in the shift register for each element in one period of the sequence. Let h(X) be the polynomial of degree k corresponding to the feedback connection of a k-stage shift register device; if h(X) is a primitive polynmial, the period of the sequence is exactly q^k-1. The set of all possible output sequences of such a register has dimension k, and therefore there are q^k such outputs with one sequence consisting totally of 0's. In the set of all k-component vectors, every field element appears in 1/q of the kq positions, that is, in kq^{k-1} positions. If the 0 vector is omitted, 0 appears only $k(q^{k-1}-1)$ times. Every nonzero element appears in one period of an m-sequence q^{k-1} times, and 0 appears $q^{k-1}-1$ times. This completely determines the distance structure of m-sequences. In the binary case where q=2 and k=m, the maximum period or length of binary m-sequences generated by a binary primitive polynomial h(x) with degree m is 2^m-1. It has been described in last section that the minimal polynomial of a primitive element used in defining $GF(2^m)$ is a primitive polynomial of degree m and with exponent $n=2^m-1$. Hence with the minimal polynomial of a primitive element of $GF(2^m)$, m-sequences over GF(2) can be generated.

2.3 PROPERTIES OF BINARY M-SEQUENCES

Binary m-sequences have been studied very extensively [5][14][8] and they have many unique properties, each of these properties especially useful in communication or ranging, as well as spread spectrum systems. Properties held by all binary m-sequences generated by a primitive polynomial h(x) with degree m, are briefly these:

P1: The number of ones in an m-sequence equals the number of zeros within one bit. This is known as the balance property of binary m-sequences. When modulating a carrier with a code sequence, one-zero balance can limit the degree of carrier suppression obtainable because carrier suppression depends on the symmetry of the modulating signal. Therefore the longer the code sequence, the less the effect of the code sequence on carrier balance.

P2: The statistical distribution of ones and zeros is well defined and always the same. If a run is defined as a series of ones or zeros grouped consecutively in a sequence; among the runs of ones and zeros in each period of an m-sequence, one half of the runs of each kind are of length one, one quarter of each kind are of length two, one eighth of each kind are of length three, and so on as long as these fractions give

meaningful numbers of runs. That is, the number of runs of each length is a decreasing power of 2 as the run length increases. When the number of stages, m, of an m-sequence is large, the sequence itself exhibits excellent randomness properties. It has been used in applications requiring random number generation and also in applications of spread spectrum systems which require pseudorandom binary sequences.

P3: If a period of the sequence is compared, term by term, with any cyclic shift of itself, the number of agreements differs from the number of disagreements by at most one. This is referred to as the correlation property of binary m-sequences. The auto-correlation function of binary m-sequences of length $n=2^m-1$ is given by

$$\theta(j) = \begin{cases} 1, & j=0 \\ -1/n, & 1 \le j < 2^m-1 . \end{cases} \qquad (2.3.1)$$

This is the best possible auto-correlation function of any binary sequence of length $n=2^m-1$, in the sense of minimizing $\max_{0<j<n} \theta(j)$.

P4: Binary m-sequences possess another valuable property. A modulo-2 addition of an m-sequence with a phase

shifted replica of itself results in another replica with a phase shift different from either of the originals.

P5: In a binary m-sequence of length $n=2^m-1$, every m consecutive elements form a unique m-tuple, each m-tuple appearing exactly once. (The all 0's m-sequence is not counted.) The m-tuples may be employed to control a processor such as a frequency synthesizer or a Monte Carlo test generator.

P6: The weight of the nonzero code word V(X) in any binary m-sequence of length $n=2^m-1$ is given by

$$|V(X)| = 2^{m-1} = (n+1)/2 \ . \tag{2.3.2}$$

If N(w) denotes the number of sequences which have weight w, the weight distribution of binary m-sequences is

$$\left.\begin{aligned} N(0)&=1 \\ N\left(\tfrac{n+1}{2}\right)&=n. \end{aligned}\right\} \tag{2.3.3}$$

P7: Let $V_1(X)$ be a binary m-sequence of length $n=2^m-1$ and d be a positive integer. Assume $V_1(X)$ is generated by the minimal polynomial $m_1(X)$ of X^n+1. A code polynomial $V_d(X)$ is obtained by taking every (d mod n)th bit of $V_1(X)$ is said to be a decimation by

d of $V_1(X)$. $V_d(X)$ has period n/GCD(n,d), and is generated by the polynomial $m_d(X)$ whose roots are the dth powers of roots of $m_1(X)$. If GCD(n,d)=1, the code V_d has length n and the decimation is called a proper decimation in that case, V_d is also an m-sequence.

Chapter III

FACTORIZATION OF BINARY PRIMITIVE POLYNOMIAL OVER GF(q)

In this chapter, methods of factoring binary primitive irreducible polynomials over $GF(q=2^s)$ are illustrated. At first, the general approach of factoring binary primitive irreducible polynomials over GF(q) will be introduced. Illustrations will be given for factoring polynomials with different degrees over various Galois fields. Special attention will be paid with respect to the $GF(q^2)$ case, for which a special result will be derived.

3.1 GENERAL APPROACH

The methodical way that one uses for factoring binary primitive polynomial over $GF(q=2^s)$ employs the knowledge of finite field theory and minimal polynomials. Which was discussed earlier in Chapter II for this purpose.

At first one needs to find a binary primitive polynomial, f(x), which is a factor of x^n+1, with degree m and exponent $n=2^m-1$. One then attempts to factor f(x) over $GF(2^s)$, where s is any positive integer and s divides m. According to the table of binary irreducible polynomials provided by Peterson & Weldon[14,Appendix C], one chooses the first entry to

calculate f(x), which implies that f(x) is a primitive polynomial ensuring that f(x) has degree m and exponent n. Table 3.1 shows some of these binary primitive irreducible polynomials, which are obtained by using the method mentioned in Chapter II, for different values of m.

m	$n=2^m-1$	f(x)
2	3	$1+x+x^2$
4	15	$1+x+x^4$
6	63	$1+x+x^6$
8	255	$1+x^2+x^3+x^4+x^8$
9	511	$1+x^4+x^9$
10	1023	$1+x^3+x^{10}$
12	4095	$1+x+x^4+x^6+x^{12}$

Table 3.1: Binary primitive irreducible factors of x^n+1, f(x), with degree m and exponent n, and $n=2^m-1$.

For $n=2^m-1$, the polynomial $f(x)$ has degree m and can be expressed as

$$f(x) = x^m+f_{m-1}x^{m-1}+f_{m-2}x^{m-2}+\ldots+f_1x+1 \ , \qquad (3.1.1)$$

where all coefficients are either 0 or 1. The next operation is to factor $f(x)$ over $GF(q=2^s)$.

Let γ be a primitive element of $GF(q)$, i.e., $GF(q)$ contains q elements and γ is an element of order $q-1$. This implies that the first $q-1$ powers of γ are distinct and hence that they include all nonzero field elements of $GF(q)$. Thus, with γ as a primitive element, $GF(q)$ can be expressed as

$$GF(q) = \{0,1,\gamma,\gamma^2,\gamma^3, \ \ldots \ ,\gamma^{q-2}\} \qquad (3.1.2)$$

and

$$\gamma^{q-1} = 1 \ . \qquad (3.1.3)$$

$GF(q)$ is treated as the ground field. The corresponding extension field can then be expressed by assigning α as a primitive element of $GF(q^k)$, where $k=m/s$, as

$$GF(q^k) = \{0,1,\alpha,\alpha^2,\alpha^3, \ \ldots \ ,\alpha^{q^k-2}\} \qquad (3.1.4)$$

$$= \{0,1,\alpha,\alpha^2,\alpha^3, \ \ldots \ ,\alpha^{2^m-2}\} \ ,$$

and

$$\alpha^{2^m-1} = 1 \ . \qquad (3.1.5)$$

Since α is the primitive element of the extension field and α is a root of $f(x)$, the logic provided for constructing the corresponding extension field is given by

$$f(\alpha) = 0 \quad \Rightarrow$$
$$\alpha^m = f_{m-1}\alpha^{m-1}+f_{m-2}\alpha^{m-2}+\ldots+f_1\alpha+1 \ . \qquad (3.1.6)$$

Therefore every element of the extension field can be expressed in terms of these m basis elements: $1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{m-1}$.

The next step is to determine how many factors of f(x) there are in GF(q). Since f(x) has degree m and with α as a primitive element of $GF(q^k)$, where $q=2^s$ and $ks=m$, this implies that there are s factors of f(x) over GF(q) and each factor is with degree k. For example, let f(x) be a primitive polynomial with degree m=6. There are two factors of f(x) over GF(4) and each factor is of degree three. However, there are three factors of f(x) over GF(8) and each factor is with degree two. Therefore, f(x) can be written in the form of,

$$f(x)=F_1(x)F_2(x)F_3(x)\ldots F_s(x), \qquad (3.1.7)$$

where $F_i(x)$'s are polynomials with degree k,

$$F_i(x)=f_{i0}+f_{i1}x+f_{i2}x^2+\ldots+f_{ik}x^k \ . \qquad (3.1.8)$$

for $1\le i\le s$ and for all f_{ij}, $0\le j\le k$. f_{ij}'s are first found as combinations of powers of α, and then they are

manipulated as elements of GF(q). Since α is a root of a binary primitive polynomial f(x), $\alpha, \alpha^2, \alpha^4, \alpha^8, \ldots, \alpha^{2^{m-1}}$ are among the m roots of f(x). The number of roots contained in each factor of f(x) is k. Table 3.2 shows the corresponding roots in each factor of f(x).

$F_i(x)$	Roots of $F_i(x)$
$F_1(x)$	$\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{k-1}}$
$F_2(x)$	$\alpha^2, \alpha^{2q}, \alpha^{2q^2}, \ldots, \alpha^{2q^{k-1}}$
$F_3(x)$	$\alpha^4, \alpha^{4q}, \alpha^{4q^2}, \ldots, \alpha^{4q^{k-1}}$
. . .	. . .
$F_i(x)$	$\alpha^{2^{i-1}}, \alpha^{2^{i-1}q}, \alpha^{2^{i-1}q^2}, \ldots, \alpha^{2^{i-1}q^{k-1}}$
. . .	. . .
$F_s(x)$	$\alpha^{q/2}, \alpha^{q^2/2}, \alpha^{q^3/2}, \ldots, \alpha^{q^k/2} (=\alpha^{2^{m-1}})$

Table 3.2: The corresponding roots of each factor of a binary primitive irreducible polynomial f(x) with degree m and exponent n.

Now each of the factors in (3.1.7) can be expressed as,

$$
\begin{aligned}
F_1(x) &= (x+\alpha)(x+\alpha^q)(x+\alpha^{q^2})\ldots(x+\alpha^{q^{k-1}}) \\
F_2(x) &= (x+\alpha^2)(x+\alpha^{2q})(x+\alpha^{2q^2})\ldots(x+\alpha^{2q^{k-1}}) \\
F_3(x) &= (x+\alpha^4)(x+\alpha^{4q})(x+\alpha^{4q^2})\ldots(x+\alpha^{4q^{k-1}}) \\
&\vdots \\
F_i(x) &= (x+\alpha^{2^{i-1}})(x+\alpha^{2^{i-1}q})(x+\alpha^{2^{i-1}q^2})\ldots(x+\alpha^{2^{i-1}q^{k-1}}) \\
&\vdots \\
F_s(x) &= (x+\alpha^{q/2})(x+\alpha^{q^2/2})(x+\alpha^{q^3/2})\ldots \\
&\quad (x+\alpha^{q^{k-1}/2})(x+\alpha^{q^k/2}) \quad .
\end{aligned}
\tag{3.1.9}
$$

Since one is factoring f(x) over GF(q), it is necessary to find out the relationship between the primitive element of GF(q), γ, and the primitive element of GF(q^k), α. From equations (3.1.3) and (3.1.4), the following can be obtained,

$$\alpha^{q^k-1} = \alpha^n = 1 \quad . \tag{3.1.10}$$

Since $2^m=q^k$, and $n=2^m-1=q^k-1$,

$$\gamma^{q-1} = \alpha^{q^k-1} \quad . \tag{3.1.11}$$

Since q-1 divides q^k-1, one essentially has that

$$\gamma = \alpha^{n/(q-1)} \quad . \tag{3.1.12}$$

After finding out the expression of γ in terms of α, one simply needs to expand each of the factors of equation (3.1.9) into polynomial form. One then has s polynomials and each polynomial is with degree k. The coefficients of each polynomial are combinations of different powers of α. One can simplify these coefficients, from the logic provided by equation (3.1.6), to some combination of powers of α which are less than m; that is, all coefficients are in the form of some combination of these basis elements of $GF(q^k)$: $1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{m-1}$. For large m, one can manipulate the expressions tactfully rather than writing down all expressions for all powers of α ; which will be shown by some examples in next section.

As shown by equation (3.1.12), one can express γ in terms of some powers of α , and thus one can express all elements in GF(q) in terms of some combinations of powers of α. Taking a look back at polynomials $F_i(x)$'s, one can compare their coefficients (in the form of combinations of powers of α) with the expressions of elements of GF(q) (also in the form of some powers of α), these coefficients can then be written by using elements from GF(q).

As long as one finds out the first factor $F_1(x)$ with coefficients belonging to GF(q), it is not necessary to use the same procedures for $F_2(x), F_3(x), \ldots, F_s(x)$. As can be observed from equation (3.1.9), all roots of $F_2(x)$ are

double those roots in $F_1(x)$, all roots of $F_3(x)$ are four-times those roots of $F_1(x)$, and so forth. Let $F_1(x)$ be of the following form:

$$F_1(x) = (x+\alpha_1)(x+\alpha_2)\ldots(x+\alpha_k) \quad , \tag{3.1.13}$$

then

$$\begin{aligned}
F_2(x) &= (x+\alpha_1^2)(x+\alpha_2^2)\ldots(x+\alpha_k^2) \\
F_3(x) &= (x+\alpha_1^4)(x+\alpha_2^4)\ldots(x+\alpha_k^4) \\
&\vdots \\
F_i(x) &= (x+\alpha_1^{2^{i-1}})(x+\alpha_2^{2^{i-1}})\ldots(x+\alpha_k^{2^{i-1}}) \\
&\vdots \\
F_s(x) &= (x+\alpha_1^{2^{s-1}})(x+\alpha_2^{2^{s-1}})\ldots(x+\alpha_k^{2^{s-1}}) \quad .
\end{aligned} \tag{3.1.14}$$

From basic algebra theory, if one expands each of the factors in (3.1.14), the following can be obtained:

$$\begin{aligned}
F_1(x) &= x^k + x^{k-1}\Big(\sum_i \alpha_i\Big) + x^{k-2}\Big(\sum_{\substack{i,j \\ i\neq j}} \alpha_i\alpha_j\Big) + \\
&\quad x^{k-3}\Big(\sum_{\substack{i,j,s \\ i\neq j\neq s}} \alpha_i\alpha_j\alpha_s\Big) + \ldots + \prod_i \alpha_i \\
&= x^k + x^{k-1}(\alpha_1+\alpha_2+\ldots+\alpha_k) + x^{k-2}(\alpha_1\alpha_2+\alpha_1\alpha_3+\ldots+\alpha_{k-1}\alpha_k) \\
&\quad +\ldots+\alpha_1\alpha_2\ldots\alpha_k
\end{aligned}$$

$$\begin{aligned}
F_2(x) &= x^k + x^{k-1}(\alpha_1^2+\alpha_2^2+\ldots+\alpha_k^2) + x^{k-2}(\alpha_1^2\alpha_2^2+\alpha_1^2\alpha_3^2+\ldots \\
&\quad +\alpha_{k-1}^2\alpha_k^2) + \ldots + \alpha_1^2\alpha_2^2\ldots\alpha_k^2
\end{aligned}$$

$$F_3(x) = x^k + x^{k-1}(\alpha_1^4 + \alpha_2^4 + \ldots + \alpha_k^4) + x^{k-2}(\alpha_1^4\alpha_2^4 + \alpha_1^4\alpha_3^4 + \ldots + \alpha_{k-1}^4\alpha_k^4) + \ldots + \alpha_1^4\alpha_2^4\ldots\alpha_k^4$$

$$\vdots$$

$$F_s(x) = x^k + x^{k-1}(\alpha_1^{2^{s-1}} + \alpha_2^{2^{s-1}} + \ldots + \alpha_k^{2^{s-1}}) + x^{k-2}(\alpha_1^{2^{s-1}}\alpha_2^{2^{s-1}} + \ldots + \alpha_{k-1}^{2^{s-1}}\alpha_k^{2^{s-1}}) + \ldots + \alpha_1^{2^{s-1}}\alpha_2^{2^{s-1}}\ldots\alpha_k^{2^{s-1}} \qquad (3.1.15)$$

It is obvious that under the modulo-2 operation, all coefficients of $F_i(x)$'s have relationships as follow,

$$\begin{aligned} f_{2j} &= (f_{1j})^2 \\ f_{3j} &= (f_{1j})^4 \\ f_{4j} &= (f_{1j})^8 \\ &\vdots \\ f_{ij} &= (f_{1j})^{2^{i-1}} \\ &\vdots \\ f_{sj} &= (f_{1j})^{2^{s-1}} , \end{aligned} \qquad (3.1.16)$$

Therefore, once one finds the expression for $F_1(x)$ with coefficients from GF(q), $F_2(x)$, $F_3(x)$, ..., and $F_s(x)$ can be obtained by simply using equation (3.1.16) since f_{ij}'s are known. As can be observed from (3.1.15), the constant term of the first factor $F(x)$ is the product of all its k roots. From Table 3.2, this constant term can then be expressed as,

$$\begin{aligned}\prod_i \alpha_i &= \alpha\ \alpha^q\ \alpha^{q^2}\ \alpha^{q^3}\ \ldots\ \alpha^{q^{k-1}} \\ &= \alpha^{(1+q+q^2+q^3+\ldots+q^{k-1})} \\ &= \alpha^{(1-q^k)/(1-q)} \\ &= \alpha^{(q^k-1)/(q-1)} \\ &= \alpha^{n/(q-1)} \\ &= \gamma \quad .\end{aligned} \tag{3.1.17}$$

Hence it can be concluded that the constant term of the first factor $F_1(x)$ always equals γ.

This is the general approach for factoring binary primitive irreducible polynomials into factors with coefficients from GF(q). It requires certain manipulation and when it comes to real examples, the calculations can be simplified and thus become easier.

3.2 SOME EXAMPLES

In this section, examples are given to illustrate the factoring of binary primitive irreducible polynomials over GF(q), with $q=2^s$. First, factoring $1+x+x^4$ over GF(4) with length n=15 will be illustrated. Then in another example with length n=63, $1+x+x^4$ will be factored first over GF(4) and then over GF(8) to indicate the differences. The method is the same for all n and q. Therefore, the different results of factors over various GF(q)'s will be tabulated rather than giving all calculation procedures.

Example 3.2.1:

In the first example, m=4 and q=4 are chosen, that is

$$n=2^4-1=15 \quad \text{and} \qquad (3.2.1)$$

$$q=2^2=4.$$

From Table 3.1, one can write down a binary primitive irreducible polynomial f(x) with degree 4 and exponent 15, as

$$f(x)=1+x+x^4 \ . \qquad (3.2.2)$$

Assuming γ is the primitive element of GF(4), and according to equation (3.1.2), one has

$$GF(4)=\{0,1,\ \gamma,\gamma^2\} \quad , \ \gamma^3=1. \qquad (3.2.3)$$

One now wants to factor $f(x)=1+x+x^4$ over GF(4), which is referred to as the ground field. The corresponding extension field $GF(4^k)$, where $k=m/s=4/2=2$, with α as the primitive element, can be expressed by using equation (3.1.4) as,

$$GF(4^2)=\{0,1,\alpha,\alpha^2,\alpha^3,\ldots,\alpha^{14}\}\ , \tag{3.2.4}$$

and $\alpha^{15}=1$. Since α is a root of $f(x)$, from equation (3.2.2), one has

$$\begin{aligned} &f(\alpha) = 1 + \alpha + \alpha^4 = 0 \quad \Rightarrow \\ &\alpha^4 = \alpha + 1\ . \end{aligned} \tag{3.2.5}$$

Equation (3.2.5) is the logic for generating $GF(4^2)$.

From equation (3.1.7), there are $s=2$ factors of $f(x)$ over GF(4) and each factor has degree $k=2$. Hence,

$$f(x)=1+x+x^4=F_1(x)F_2(x), \tag{3.2.6}$$

where

$$F_1(x) = f_{12}x^2 + f_{11}x + f_{10} \tag{3.2.7}$$

and

$$F_2(x) = f_{22}x^2 + f_{21}x + f_{20}\ . \tag{3.2.8}$$

Since α is a root of $f(x)$, α , α^2, α^4, α^8 are among all roots of $f(x)$. According to Table 3.2,

$$F_1(x)=(x+\alpha)(x+\alpha^4)$$
$$F_2(x)=(x+\alpha^2)(x+\alpha^8). \qquad (3.2.9)$$

As mentioned in section above, one needs only to find out the corresponding coefficients of $F_1(x)$, so $F_1(x)$ can be expanded into polynomial form as

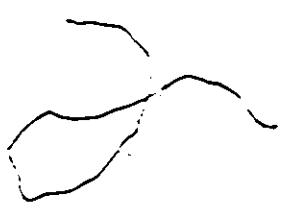
$$F_1(x)=f_{12}x^2+f_{11}x+f_{10}$$
$$= x^2 + (\alpha+\alpha^4)x + \alpha^5 \quad . \qquad (3.2.10)$$

Now one wants to simplify all coefficients of $F_1(x)$ in terms of 1, α, α^2, and α^3 by using the logic provided by equation (3.2.5). The following are obtained:

$$f_{12} = 1 \ ,$$
$$f_{11} = \alpha + \alpha^4 = \alpha + \alpha + 1 = 1,$$
$$f_{10} = \alpha^5 = \alpha(\alpha + 1) = \alpha^2 + \alpha \quad . \qquad (3.2.11)$$

The next step is to relate γ , the primitive element of GF(4), with α , the primitive element of $GF(4^2)$. From equation (3.1.12),

$$\gamma = \alpha^{15/3} = \alpha^5 \ . \qquad (3.2.12)$$



And the elements of GF(4) can be expressed in terms of powers of α as:

$$\begin{aligned} \gamma &= \alpha^5 \ , \\ \gamma^2 &= \alpha^{10} \ , \\ \gamma^3 &= \alpha^{15} = 1 \ . \end{aligned} \tag{3.2.13}$$

In order to simplify equation (3.2.13), one needs to calculate the following,

$$\begin{aligned} \alpha^8 &= (\alpha^4)^2 = \alpha^2 + 1 \ , \\ \alpha^{10} &= \alpha^8\alpha^2 = \alpha^2(\alpha^2+1) = \alpha^2 + \alpha + 1 \ . \end{aligned} \tag{3.2.14}$$

Therefore, equation (3.2.13) becomes

$$\begin{aligned} \gamma &= \alpha^5 = \alpha^2 + \alpha \ , \\ \gamma^2 &= \alpha^{10} = \alpha^2 + \alpha + 1 \ , \\ \gamma^3 &= \alpha^{15} = 1 \ . \end{aligned} \tag{3.2.15}$$

By observing equation (3.2.15), one can immediately write down the logic for GF(4), which is,

$$\gamma^2 = \gamma + 1 \ . \tag{3.2.16}$$

And then one can compare (3.2.11) with (3.2.15) and the coefficients of $F_1(x)$ can be expressed by elements from GF(4). That is,

$$\begin{aligned} f_{12} &= 1 \ , \\ f_{11} &= 1 \ , \\ f_{10} &= \alpha^5 = \gamma \ . \end{aligned} \tag{3.2.17}$$

Thus $F_1(x)$ can be expessed by elements of GF(4) as

$$F_1(x) = x^2 + x + \gamma \quad . \tag{3.2.18}$$

This also verifies equation (3.1.17). By employing equation (3.1.16), the coefficients of $F_2(x)$ can be written as

$$\begin{aligned} f_{22} &= (f_{12})^2 = 1 \quad , \\ f_{21} &= (f_{11})^2 = 1 \quad , \\ f_{20} &= (f_{10})^2 = \gamma^2 \quad . \end{aligned} \tag{3.2.19}$$

Hence $$F_2(x) = x^2 + x + \gamma^2 \quad . \tag{3.2.20}$$

Consequently, f(x) can be factored as

$$\begin{aligned} f(x) &= x^4 + x + 1 \\ &= (x^2+x+\gamma)(x^2+x+\gamma^2) \quad . \end{aligned} \tag{3.2.21}$$

over GF(4) with $\gamma^2=\gamma+1$ as the logic for GF(4). For the sake of argument, find $F_2(x)$ by the same approach used to find $F_1(x)$. First expand $F_2(x)$ as

$$F_2(x) = x^2 + (\alpha^2+\alpha^8)x + \alpha^{10} \quad . \tag{3.2.22}$$

The coefficients of $F_2(x)$ thus become

$$\begin{aligned} f_{21} &= \alpha^2+\alpha^8 = \alpha^2+ \alpha^2+ 1 = 1 \quad , \\ f_{20} &= \alpha^{10} = \alpha^2 +\alpha + 1 =\gamma^2 \quad . \end{aligned} \tag{3.2.23}$$

$F_2(x)=x^2+x+\gamma^2$, is exactly the same as equation (3.2.20), verifying that equation (3.1.16) is valid. As a last step,

verify equation (3.2.21). The right hand side of equation (3.2.21) can be expanded to

$$\begin{aligned} & (x^2+x+\gamma)(x^2+x+\gamma^2) \\ &= x^4+x^3(1+1)+x^2(\gamma^2+\gamma+1)+x(\gamma^2+\gamma)+\gamma^3 \\ &= x^4+x+1 \\ &= f(x). \end{aligned} \tag{3.2.24}$$

Therefore the example is completed and it can be concluded that equation (3.2.21) is the answer for factoring $1+x+x^4$ over GF(4).

From this simple example, it is seen that by careful observation and manipulation, it isn't necessary to calculate all expressions of all powers of α. It is only necessary to calculate one of the factors of f(x). Thus, factoring binary primitive irreducible polynomial over GF(q) is not as complicated as one might think. Also it is to be noticed that the logic for GF(q) arises from its extension field $GF(q^k)$ rather than being assigned to GF(q).

Example 3.2.2:

In this example, m=6 and q=4, that is, $n=2^6-1=63$ and $q=2^2=4$. The binary primitive irreducible polynomial taken from Table 3.1 is,

$$f(x)=1+x+x^6 . \tag{3.2.25}$$

With q=4, the ground field GF(4) in this example is,

$$GF(4)=\{0,1,\gamma,\gamma^2\},\ \gamma^3 = 1\ , \tag{3.2.26}$$

where γ is the primitive element of GF(4). The corresponding extension field, $GF(4^3)$, with α as the primitive element, can be written as,

$$GF(4^3)=\{0,1,\alpha,\alpha^2,\ \ldots\ ,\alpha^{62}\}\ ,\ \alpha^{63} = 1\ . \tag{3.2.27}$$

Since α is a root of f(x), the logic for $GF(4^3)$ is

$$\alpha^6 = \alpha + 1\ . \tag{3.2.28}$$

In this case f(x) has two factors over GF(4) and each factor is with degree three. That is,

$$f(x) = F_1(x)\ F_2(x)\ ,$$

and

$$F_1(x) = f_{13}x^3 + f_{12}x^2 + f_{11}x + f_{10}\ ,$$

$$F_2(x) = f_{23}x^3 + f_{22}x^2 + f_{21}x + f_{20}\ .$$

Among all roots of f(x) are: α , α^2, α^4, α^8, α^{16}, α^{32}, and these are distributed to $F_1(x)$ and $F_2(x)$ according to Table 3.2 as:

$$F_1(x) = (x+\alpha)\ (x+\alpha^4)\ (x+\alpha^{16})\ ,$$

$$F_2(x) = (x+\alpha^2)\ (x+\alpha^8)\ (x+\alpha^{32})\ . \tag{3.2.29}$$

Taking the first factor $F_1(x)$ and expanding it, the result is

$$F_1(x) = f_{13}x^3 + f_{12}x^2 + f_{11}x + f_{10} \qquad (3.2.30)$$
$$= x^3 + (\alpha+\alpha^4+\alpha^{16})x^2 + (\alpha^5+\alpha^{17}+\alpha^{20})x + \alpha^{21} .$$

Observing the coefficients of $F_1(x)$, and employing the logic given by equation (3.2.28), the following results are reached.

$$\alpha^6 = \alpha + 1$$
$$\alpha^6 = \alpha^3 + \alpha^2$$
$$\alpha^{10} = \alpha^5 + \alpha^4$$
$$\alpha^{12} = (\alpha^6)^2 = \alpha^2 + 1$$
$$\alpha^{16} = \alpha^4 + \alpha + 1$$
$$\alpha^{17} = \alpha^5 + \alpha^2 + \alpha$$
$$\alpha^{20} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2$$
$$\alpha^{21} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1$$
$$\alpha^{40} = (\alpha^{20})^2 = \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1 \quad . \qquad (3.2.31)$$

Substitute (3.2.31) into (3.2.30),

$$F_1(x) = x^3 + x^2 + (\alpha^5+\alpha^4+\alpha^3+\alpha)x + \alpha^{21}. \qquad (3.2.32)$$

γ can now be related to α as

$$\gamma = \alpha^{63/3} = \alpha^{21} , \qquad (3.2.33)$$

and

$$\gamma = \alpha^{21} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1 \quad ,$$

$$\gamma^2 = \alpha^{42} = \alpha^2(\alpha^{40}) = \alpha^5 + \alpha^4 + \alpha^3 + \alpha \quad ,$$

$$\rightarrow \quad \gamma^2 = \gamma + 1 \quad . \tag{3.2.34}$$

Equation (3.2.34) gives the logic for GF(4). Comparing equation (3.2.32) with equation (3.2.34), the result is

$$F_1(x) = x^3 + x^2 + \gamma^2 x + \gamma \quad . \tag{3.2.35}$$

The coefficients of $F_2(x)$ can then be written as,

$$\begin{aligned} f_{23} &= (f_{13})^2 = 1 \quad , \\ f_{22} &= (f_{12})^2 = 1 \quad , \\ f_{21} &= (f_{11})^2 = \gamma^4 = \gamma \quad , \\ f_{20} &= (f_{10})^2 = \gamma^2 \quad . \end{aligned} \tag{3.2.36}$$

Therefore,

$$F_2(x) = x^3 + x^2 + \gamma x + \gamma^2 \quad . \tag{3.2.37}$$

And f(x) can be factored as:

$$\begin{aligned} f(x) &= x^6 + x + 1 \\ &= (x^3+x^2+\gamma^2 x+\gamma)\ (x^3+x^2+\gamma x+\gamma^2) \end{aligned} \tag{3.2.38}$$

with coefficients over GF(4) and logic provided by $\gamma^2 = \gamma + 1$. To verify equation (3.2.38), the right hand side of this equation can be written as,

$$(x^3+x^2+\gamma^2x+\gamma)(x^3+x^2+\gamma x+\gamma^2)$$

$$= x^6+x^5(1+1)+x^4(\gamma+1+\gamma^2)+x^3(\gamma^2+\gamma+\gamma^2+\gamma)$$

$$+x^2(\gamma^2+1+\gamma)+x(\gamma+\gamma^2)+\gamma^3$$

$$= x^6+x+1$$

$$= f(x) \quad . \tag{3.2.39}$$

This completes the factorization of $1+x+x^6$ over GF(4).

Example 3.2.3:

In this example, the same binary primitive irreducible polynomial $f(x)=1+x+x^6$ is used as in example 3.2.2; but it is factored over GF(8) instead of GF(4). That is, $m=6$ and $n=2^6-1=63$ again, but $q=2^3=8$. Let γ be the primitive element of GF(8),

$$GF(8) = \{0, 1, \gamma, \gamma^2, \ldots, \gamma^6\}, \quad \gamma^7 = 1 \quad . \tag{3.2.40}$$

The corresponding extension field $GF(8^2)$, with α as the primitive element, is given by

$$GF(8^2)=\{0,1,\alpha,\alpha^2, \ldots ,\alpha^{62}\}, \quad \alpha^{63} = 1 \quad . \tag{3.2.41}$$

Since α is a root of f(x), the logic of $GF(8^2)$ is given by

$$\alpha^6 = \alpha + 1 \quad . \tag{3.2.42}$$

As in the previous example, among all roots of f(x) are α, α^2, α^4, α^8, α^{16} and α^{32}, but in this case there are three

factors of f(x) over GF(8) and each of these factors is with degree two. That is,

$$f(x)=F_1(x)F_2(x)\overline{F_3(x)} \ .$$

Let

$$\begin{aligned} F_1(x) &= f_{12}x^2 + f_{11}x + f_{10} \ , \\ F_2(x) &= f_{22}x^2 + f_{21}x + f_{20} \ , \\ F_3(x) &= f_{32}x^2 + f_{31}x + f_{30} \ . \end{aligned} \tag{3.2.43}$$

Then each of the factors can be expressed as,

$$\begin{aligned} F_1(x) &= (x+\alpha)(x+\alpha^8) = x^2 + (\alpha+\alpha^8)x + \alpha^9 \ , \\ F_2(x) &= (x+\alpha^2)(x+\alpha^{16}) = x^2 + (\alpha^2+\alpha^{16})x + \alpha^{18}, \\ F_3(x) &= (x+\alpha^4)(x+\alpha^{32}) = x^2 + (\alpha^4+\alpha^{32})x + \alpha^{36}. \end{aligned} \tag{3.2.44}$$

Also given is

$$\gamma = \alpha^{63/7} = \alpha^9 \ . \tag{3.2.45}$$

From the logic provided by (3.2.42), the coefficients of $F_1(x)$ and elements from GF(8) can be expressed in terms of the basis elements 1, α, α^2, α^3, α^4 and α^5. Thus obtaining

$$\begin{aligned} F_1(x) &= x^2 + (\alpha+\alpha^8)x + \alpha^9 \\ &= x^2 + (\alpha+\alpha^3+\alpha^2)x + \alpha^9 \\ &= x^2 + \gamma^3 x + \gamma \ . \end{aligned} \tag{3.2.46}$$

And the logic provided for GF(8), arising from the logic provided for GF(8^2) by (3.2.42), is

$$\gamma^3 = \gamma^2 + 1 \quad . \tag{3.2.47}$$

Since $F_1(x)$ has been expressed in terms of elements from GF(8), $F_2(x)$ and $F_3(x)$ can then be written as

$$\begin{aligned} F_2(x) &= x^2 + \gamma^6 x + \gamma^2 \ , \\ F_3(x) &= x^2 + \gamma^5 x + \gamma^4 \ . \end{aligned} \tag{3.2.48}$$

Therefore,

$$\begin{aligned} f(x) &= x^6 + x + 1 \\ &= (x^2+\gamma^3 x+\gamma)(x^2 + \gamma^6 x +\gamma^2)(x^2 + \gamma^5 x + \gamma^4) \end{aligned} \tag{3.2.49}$$

with coefficients over GF(8) where $\gamma^3=\gamma^2+1$. To verify the above factorization, express the right hand side of (3.2.49) as:

$$\begin{aligned} & (x^2+\gamma^3 x+\gamma)(x^2+\gamma^6 x+\gamma^2)(x^2+\gamma^5 x+\gamma^4) \\ &= x^6+x^5(\gamma^5+\gamma^6+\gamma^3)+x^4(\gamma^2+\gamma^9+\gamma+\gamma^{11}+\gamma^8+\gamma^4) \\ &\quad +x^3(\gamma^7+\gamma^{14}+\gamma^6+\gamma^{10}+\gamma^7+\gamma^5+\gamma^7)+x^2(\gamma^6+\gamma^{13}+\gamma^5+\gamma^3+\gamma^{10}+\gamma^{12}) \\ &\quad +x(\gamma^8+\gamma^9+\gamma^{11})+\gamma^7 \\ &= x^6+x+1 \\ &= f(x) \ . \end{aligned}$$

Hence the factorization is completed and equation (3.2.49) is the exact answer for factoring $1+x+x^4$ over GF(8) with logic provided by $\gamma^3=\gamma^2+1$.

Three examples illustrating the factorization of binary primitive irreducible polynomials over $GF(q=2^s)$ have now been shown. The method discussed holds for any m and q, hence it is possible to list some results rather than showing all necessary steps. Table 3.3 shows the factors of binary primitive irreducible polynomials for different values of m and/or of different Galois fields.

n $=2^m-1$	q $=2^s$	$f(x)$, a binary irr. factor of x^n+1 with exp. n	γ $=\alpha^{n/(q-1)}$	logic for GF(q)	Factors of $f(x)$ over GF(q)
15	4	x^4+x+1	$\gamma=\alpha^5$	$\gamma^2=\gamma+1$	$(x^2+x+\gamma)(x^2+x+\gamma^2)$
63	4	x^6+x+1	$\gamma=\alpha^{21}$	$\gamma^2=\gamma+1$	$(x^3+x^2+\gamma^2x+\gamma)(x^3+x^2+\gamma x+\gamma^2)$
	8		$\gamma=\alpha^9$	$\gamma^3=\gamma^2+1$	$(x^2+\gamma^3x+\gamma)(x^2+\gamma^6x+\gamma^2)(x^2+\gamma^5x+\gamma^4)$
255	4	$x^8+x^4+x^3+x^2+1$	$\gamma=\alpha^{85}$	$\gamma^2=\gamma+1$	$(x^4+x^3+\gamma x^2+\gamma x+\gamma)(x^4+x^3+\gamma^2x^2+\gamma^2x+\gamma^2)$
	16		$\gamma=\alpha^{17}$	$\gamma^4=\gamma+1$	$(x^2+\gamma^2x+\gamma)(x^2+\gamma^4x+\gamma^2)$ $(x^2+\gamma^8x+\gamma^4)(x^2+\gamma x+\gamma^8)$
511	8	x^9+x^4+1	$\gamma=\alpha^{73}$	$\gamma^3=\gamma+1$	$(x^3+\gamma x^2+\gamma^5x+\gamma)(x^3+\gamma^2x^2+\gamma^3x+\gamma^2)$ $(x^3+\gamma^4x^2+\gamma^6x+\gamma^4)$
1023	4	$x^{10}+x^3+1$	$\gamma=\alpha^{341}$	$\gamma^2=\gamma+1$	$(x^5+x^4+\gamma^2x^3+\gamma x^2+\gamma)$ $(x^5+x^4+\gamma x^3+\gamma^2x^2+\gamma^2)$
4095	4	$x^{12}+x^6+x^4+x+1$	$\gamma=\alpha^{1365}$	$\gamma^2=\gamma+1$	$(x^6+x^2+x+\gamma)(x^6+x^2+x+\gamma^2)$
	8		$\gamma=\alpha^{585}$	$\gamma^3=\gamma+1$	$(x^4+\gamma x^3+\gamma^3x^2+x+\gamma)(x^4+\gamma^2x^3+\gamma^6x^2$ $+x+\gamma^2)(x^4+\gamma^4x^3+\gamma^5x^2+x+\gamma^4)$
	16		$\gamma=\alpha^{273}$	$\gamma^4=\gamma+1$	$(x^3+\gamma^{10}x^2+\gamma^{13}x+\gamma)(x^3+\gamma^5x^2+\gamma^{11}x+\gamma^2$ $(x^3+\gamma^{10}x^2+\gamma^7x+\gamma^4)(x^3+\gamma^5x^2+\gamma^{14}x+\gamma^8)$

Table 3.3 : Factors of binary primitive irreducible polynomials with different lengths over different Galois fields.

3.3 THE SPECIAL CASE OF GF(q^2)

In this section special attention will be given to an extension field GF(q^2) with respect to a ground field GF(q), where $q=2^s$. It is necessary to introduce the concept of 'trace' here before going further. The trace [3][11][2] of an element $\beta \in GF(p^k)$, abbreviated Tr(β), is defined by

$$\begin{aligned} Tr(\beta) &= \sum_{i=0}^{k-1} \beta^{p^i} \\ &= \beta + \beta^p + \beta^{p^2} + \ldots + \beta^{p^{k-1}} . \end{aligned} \tag{3.3.1}$$

Trace has the following properties:

$$\begin{aligned} &Tr(\beta) \in GF(p) \ , \\ &Tr(\beta+\gamma) = Tr(\beta) + Tr(\gamma) \ , \\ &Tr(\beta^p) = Tr(\beta)^p = Tr(\beta) \ , \\ &Tr(1) = k \bmod p \ , \end{aligned} \tag{3.3.2}$$

for all $\gamma, \beta \in GF(p^k)$.

Recall that if γ is the primitive element of GF(q), where $q=2^s$,

$$GF(q) = \{0, 1, \gamma, \gamma^2, \ldots, \gamma^{\nu-1}\} , \tag{3.3.3}$$

where $\nu=q-1=2^s-1$ and $\gamma^\nu=1$. The corresponding extension field, GF(q^2), with α as the primitive element, is given by

$$GF(q^2) = \{0, 1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{n-1}\} , \tag{3.3.4}$$

where $n=q^2-1=2^{2s}-1$, and $\alpha^n=1$. Consider the polynomial $g(x)$, where

$$g(x)=x^2+x+\gamma \quad , \tag{3.3.5}$$

with α as a root of $g(x)$, the following theorem is obtained.

Theorem I : $\alpha^{2^s+1} = \gamma$ iff $Tr(\gamma)=1$.

The proof of Theorem I is as follow:

Proof: If α is a root of $g(x)=x^2+x+\gamma$, then

$$\alpha^2 = \gamma + \alpha \quad . \tag{3.3.6}$$

By squaring (3.3.6) repeatedly, the following is obtained,

$$\begin{aligned}
&\alpha^4 = \alpha^2+ \gamma^2 = \gamma^2+ \gamma+ \alpha \quad , \\
&\alpha^4 = \gamma^4+ \gamma^2+ \alpha^2 = \gamma^4+ \gamma^2+ \gamma+ \alpha \ , \\
&\alpha^{16} = \gamma^8+ \gamma^4+ \gamma^2+ \alpha^2 = \gamma^8+ \gamma^4+ \gamma^2+ \gamma+ \alpha \ , \\
&\quad \vdots \\
&\alpha^{2^i} = \gamma^{2^{i-1}} + \gamma^{2^{i-2}} + \ldots + \gamma^4+ \gamma^2+ \gamma+ \alpha \ , \\
&\quad \vdots \\
&\alpha^{2^s} = \gamma^{2^{s-1}} + \gamma^{2^{s-2}} + \ldots + \gamma^4+ \gamma^2+ \gamma+ \alpha \ .
\end{aligned} \tag{3.3.7}$$

Multiplying both sides of (3.3.7) by α, gives

$$\alpha^{2^s+1} = \alpha(\gamma^{2^{s-1}}+\gamma^{2^{s-2}}+ \ldots +\gamma^2+\gamma)+\alpha^2$$

$$= \alpha(\gamma^{2^{s-1}}+\gamma^{2^{s-2}}+ \ldots +\gamma^2+\gamma+1)+\gamma \quad , \qquad (3.3.8)$$

which means that

$$\alpha^{2^s+1} = \gamma \quad \text{iff}$$

$$1 + \gamma + \gamma^2 + \gamma^4 + \ldots + \gamma^{2^{s-1}} = 0 \quad , \qquad (3.3.9)$$

and consequently,

$$\alpha^{2^s+1} = \gamma \quad \text{iff} \quad Tr(\gamma) = 1 \ .$$

Q.E.D.

The specially chosen polynomial $g(x)=x^2+x+\gamma$, which has α as a root, where $\alpha \in GF(q^2)$, is indeed one factor of a binary primitive irreducible polynomial $f(x)$ with degree m and exponent n, where $n=2^m-1$. According to (3.1.7), (3.1.8) and (3.1.16), the corresponding $f(x)$ is

$$f(x) = (x^2 + x + \gamma)\ (x^2 + x + \gamma^2)\ (x^2 + x + \gamma^4)$$
$$\ldots (x^2+x+\gamma^{2^{s-2}})\ (x^2+x+\gamma^{2^{s-1}}) \ . \qquad (3.3.10)$$

With a suitably chosen logic provided for GF(q), f(x) can be determined. It is to be noticed that with that particular case in which Theorem I holds, the corresponding binary primitive irreducible polynomial f(x) is not necessarily the

first entry of the table of binary primitive polynomials provided by Peterson & Weldon[14,Appendix C]. The reason being that f(x) has one constraint, $Tr(\gamma)=1$, for $\gamma \in GF(q)$. According to equation (2.1.9), the number of binary primitive polynomials of degree m and exponent n is given by $\theta(m)=\phi(n)/m$, where $\phi(n)$ is the Euler number of n , where $n=2^m-1$. Since there are $\phi(n)/m$ such binary prmitive polynomials, the one with $x^2+x+\gamma$ as a factor may not necessarily be the one corresponding to the first entry of this table.

Table 3.4 shows some examples of binary primitive irreducible polynomials which have $x^2+x+\gamma$ as a factor where α is a root of this factor, and $Tr(\gamma)=1$, where α is the primitive element of $GF(q^2)$, $q=2^s$ and $\alpha^{2^s+1}=\gamma$. The first column of this table lists the number n where $n=2^m-1$ while the second column shows the value of q which satisfies $q^2=2^m=n+1$. The third column shows the relationship between γ and α , which is given by $\gamma = \alpha^{2^s+1}=\alpha^{q+1}$. The fourth column lists the logic chosen for the corresponding GF(q), in which it ensures that $Tr(\gamma)=1$ and has degree s. The fifth column, by using (3.3.10), shows all factors of a binary polynomial which has $x^2+x+\gamma$ as one factor. The sixth column lists such a binary primitive irreducible polynomial with degree m and exponent n. The last column indicates the corresponding primitive root of f(x) in $GF(q^2)$.

n	q	$\gamma=\alpha^{2^s+1}$	Logic for GF(q)	Factors of f(x) over GF(q)	f(x), an irr. factor of $1+x^n$ over GF(2)	The primitive element of f(x)
15	4	$\gamma=\alpha^5$	$\gamma^2=\gamma+1$	$(x^2+x+\gamma)(x^2+x+\gamma^2)$	x^4+x+1	α
63	8	$\gamma=\alpha^9$	$\gamma^3=\gamma^2+1$	$(x^2+x+\gamma)(x^2+x+\gamma^2)$ $(x^2+x+\gamma^4)$	$x^6+x^5+x^3+x^2+1$	α^{11}
255	16	$\gamma=\alpha^{17}$	$\gamma^4=\gamma^3+1$	$(x^2+x+\gamma)(x^2+x+\gamma^2)$ $(x^2+x+\gamma^4)(x^2+x+\gamma^8)$	$x^8+x^6+x^5+x^3+1$	α^7
1023	32	$\gamma=\alpha^{33}$	$\gamma^5=\gamma^4+\gamma^3+$ γ^2+1	$(x^2+x+\gamma)(x^2+x+\gamma^2)(x^2+x+\gamma^4)$ $(x^2+x+\gamma^8)(x^2+x+\gamma^{16})$	$x^{10}+x^9+x^8+x^4$ $+x^3+x^2+1$	α^{343}
4095	64	$\gamma=\alpha^{65}$	$\gamma^6=\gamma^5+1$	$(x^2+x+\gamma)(x^2+x+\gamma^2)(x^2+x+\gamma^4)$ $(x^2+x+\gamma^8)(x^2+x+\gamma^{16})$ $(x^2+x+\gamma^{32})$	$x^{12}+x^9+x^8+x^5+1$	α^{197}

Table 3.4 : Binary primitive irreducible polynomials with $x^2+x+\gamma$ as a factor over GF(q), where α is a root of $x^2+x+\gamma$ and $\alpha \in GF(q^2)$.

Chapter IV

CONSTRUCTION OF C-M-SEQUENCES OVER GF(q)

In the last chapter it was shown how to factor a binary primitive irreducible polynomial, f(x), over GF(q) with $q=2^s$ where there are s factors in every f(x) with respect to different GF(q)'s. Each of these factors is irreducible over GF(q). One of these primitive factors can be used as the corresponding parity check polynomial to construct the q-nary m-sequences. First the general approach of constructing q-nary c-m-sequences will be outlined and illustrations will be given upon various lengths of such sequences. Finally, attention will be paid to the $GF(q^2)$ case.

4.1 GENERAL APPROACH AND ILLUSTRATIONS

As mentioned before, m-sequences are sequences which have the maximum period possible for a linear feedback shift register of k stages, corresponding to a primitive polynomial of degree k. The period of the q-nary m-sequences is exactly q^k-1, and they can be generated by a k-stage shift register with feedback function determined by the primitive parity check polynomial with degree k. As for

q-nary 'characteristic' m-sequences, one constraint is added.

Definition: If an m-sequence over GF(q) is represented by an n-bit code vector, where $n=2^m-1=q^k-1$,

$$V=(v_0,v_1,v_2,\ldots,v_i,\ldots,v_{n-1}) , \qquad (4.1.1)$$

for all $0\le i\le n-1$, $v_i \in GF(q)$, and if

$$v_i = v_{qi} , \qquad (4.1.2)$$

V is called a characteristic m-sequence or simply a c-m-sequence.

Without the constraint of (4.1.2), the code may not be one-step majority logic decodable. It will be shown why the c-m-sequences are one-step majority logic decodable after the method for constructing such sequences is illustrated.

It is known that either the generator polynomial or the parity check polynomial is needed in order to generate the m-sequences. In the method described below, the parity check polynomial is needed to determine the feedback function of a feedback shift register device and thus to construct the c-m-sequences corresponding to some arbitrary chosen initial conditions.

The parity check polynomial which is to be used is one of the factors of the binary primitive irreducible polynomial f(x) with degree m and exponent n, over GF(q), where $n=2^m-1=q^k-1$; as provided by Table 3.3. The first factor, $F_1(x)$, a primitive polynomial over GF(q) with degree k, is chosen to be the parity check polynomial h(x). However, it is also possible to generate c-m-sequences by using the other factors of f(x) as the parity check polynomial, as long as the chosen factor is primitive over GF(q). The methodical approach will be used where by the initial conditions by the recursive formula defined by V·H=0 will be found, where H is the parity check matrix defined by the parity check polynomial h(x). These will then be used to construct the c-m-sequences by linear shift register devices with feedback logic determined by h(x).

The above approach is best understood by working through some examples. The examples given below illustrate the construction of c-m-sequences over GF(q). It must be realized however that once one such sequence is obtained, all q^k-1 m-sequences over GF(q) can also be obtained by simply shifting the sequence cyclically. Therefore, it is not necessary to illustrate the generation of general m-sequences over GF(q).

Example 4.1.1:

For the case when m=4, q=4, then n=2^4-1=15; and from Table 3.3, $x^2+x+\gamma$ is a factor of x^4+x+1 with α as a root, where $\alpha \in GF(4^2)$. Therefore $GF(4)=\{0,1,\gamma,\gamma^2\}$ and $\gamma^3=1$. The logic provided for GF(4) is $\gamma^2=\gamma+1$. Now $x^2+x+\gamma$ can be used as the parity check polynomial, that is,

$$h(x)=x^2+x+\gamma \ , \qquad (4.1.3)$$

where $\gamma^2=\gamma+1$. Let $V=(v_0,v_1,v_2,\ldots,v_{14})$ be the 15-bit code generated by h(x). In order to yield the c-m-sequences, $v_i=v_{qi}$ or in this case,

$$v_i = v_{4i} \ . \qquad (4.1.4)$$

Therefore the corresponding coset numbers for q=4 and n=15 are

1,4

2,8

3,12

5

6,9

7,13

10

11,14 , (4.1.5)

for later use in calculations. Since $h^*(x)$ generates the reciprocal of V, V*, starting with $h(x)=x^2+x+\gamma$, with the inner product of V and H being zero, the following equations are obtained:

$$\begin{aligned}
&\gamma v_0 + v_1 + v_2 = 0 \\
&\gamma v_1 + v_2 + v_3 = 0 \\
&\gamma v_2 + v_3 + v_1 = 0 \\
&\gamma v_3 + v_1 + v_5 = 0 \\
&\quad\vdots \\
&\gamma v_{i-2} + v_{i-1} + v_i = 0 \\
&\quad\vdots
\end{aligned} \tag{4.1.6}$$

where $\gamma^2=\gamma+1$. The purpose is to express all v_i's in terms of v_0 if $v_0\neq 0$, and in terms of v_1 if $v_0=0$. Because of the unique property given by (4.1.4), and by observing (4.1.6), if v_0 is fixed, there are only four variables namely v_1, v_2, v_3 and v_5 in the top four equations. Hence these four equations can be solved and the following results are obtained:

$$\begin{aligned}
&v_0 = 0 , \\
&v_2 = v_1 , \\
&v_3 = \gamma^2 v_1 , \\
&v_5 = 0 .
\end{aligned} \tag{4.1.7}$$

The recursive formula according to h(x) can be written as

$$v_i = v_{i-1} + \gamma v_{i-2} . \tag{4.1.8}$$

By using (4.1.7) and (4.1.8), it is possible to find all v_i's in terms of v_1, for examples,

$$\begin{aligned} v_6 &= v_5 + \gamma v_4 = \gamma v_1 \\ v_7 &= v_6 + \gamma v_5 = \gamma v_1 \\ v_8 &= v_2 = v_1 \\ v_9 &= v_8 + \gamma v_7 = \gamma v_1 \\ &\vdots \end{aligned} \tag{4.1.9}$$

The sequence repeats itself after v_{14}. According to (4.1.8), only the two initial stages need to be known and then the whole sequence of length 15 can be constructed by using (4.1.8) recursively. Since $v_0=0$, only three different initial conditions for v_1 can be assigned, namely 1, γ and γ^2. That is because the all zero sequence has been excluded and hence only three nonzero elements from GF(4) remain. With these three different initial conditions, the three different c-m-sequences can be obtained, designated by V_1, V_2 and V_3 respectively as follows:

$$\begin{aligned} V_1 &: \ 0 \ 1 \ 1 \ \gamma^2 \ 1 \ 0 \ \gamma \ \gamma \ 1 \ \gamma \ 0 \ \gamma^2 \ \gamma^2 \ \gamma \ \gamma^2 \\ V_2 &: \ 0 \ \gamma \ \gamma \ 1 \ \gamma \ 0 \ \gamma^2 \ \gamma^2 \ \gamma \ \gamma^2 \ 0 \ 1 \ 1 \ \gamma^2 \ 1 \\ V_3 &: \ 0 \ \gamma^2 \ \gamma^2 \ \gamma \ \gamma^2 \ 0 \ 1 \ 1 \ \gamma^2 \ 1 \ 0 \ \gamma \ \gamma \ 1 \ \gamma \end{aligned} \tag{4.1.10}$$

These three sequences start repeating themselves after v_{14}, showing that the period of these sequences is 15. It is also observed that the sequences are a cyclic shift of each other. For examples, if $V_1(X)$, $V_2(X)$ and $V_3(X)$ represent these three sequences corresponding to initial conditions $(v_1, v_0)=(1,0)$, $(\gamma,0)$, and $(\gamma^2,0)$ respectively, then

$$V_2(X) = X^{10} V_1(X) \bmod (X^{15}+1)$$
$$V_3(X) = X^5 V_1(X) \bmod (X^{15}+1) \quad . \tag{4.1.11}$$

Also, observing each sequence independently, it is found that $v_i=v_{4i}$ is true for all of them. Therefore, three c-m-sequences over GF(4) according to the parity check polynomial $h(x)=x^2+x+\gamma$ with $\gamma^2=\gamma+1$ have been generated. Figure 4.1(a) shows the 2-stage shift register device which can be used to generate the c-m-sequences by loading the three different initial conditions on the shift registers. In practice, since each element of GF(4) can be represented by a 2-tuple over GF(2), Figure 4.1(a) can be implemented in binary shift registers fashion, as shown in the following:

$$v_0 = a_0 + a_1\gamma$$
$$v_1 = b_0 + b_1\gamma \quad , \tag{4.1.12}$$

where all coefficients a_0, a_1, b_0 and b_1 are either 0 or 1. According to the recursive polynomial of (4.1.8), the following expression holds:

$$\begin{aligned} v_2 &= v_1 + \gamma v_0 \\ &= (b_0+b_1\gamma) + \gamma(a_0+a_1\gamma) \\ &= (a_1+b_0) + (a_0+a_1+b_1)\gamma \quad , \end{aligned} \tag{4.1.13}$$

where the logic is provided by $\gamma^2 = \gamma+1$. Figure 4.1(b) illustrates the binary implementation of Figure 4.1(a) with circuit connection in accordance with equation (4.1.13). The binary output (x,y) represents a symbol in GF(4), i.e.,

$$v_i = x+y\gamma \quad , \text{ for } 0 \le i \le n \ . \tag{4.1.14}$$

The 3 initial states to be loaded on the shift registers to generate the c-m-sequences thus become

$$\begin{aligned} (v_1,v_0) &= ((b_0,b_1),(a_0,a_1)) \\ &= \begin{cases} ((1,0),(0,0)) \\ ((0,1),(0,0)) \\ ((1,1),(0,0)) \end{cases} \ . \end{aligned} \tag{4.1.15}$$

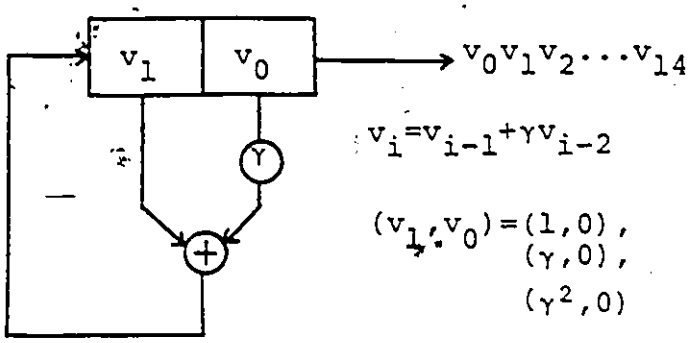


Fig. 4.1(a): A 2-stage shift register for generating the c-m-sequences over GF(4). The feedback connection is according to $h(x)=x^2+x+\gamma$, with $\gamma^2=\gamma+1$.

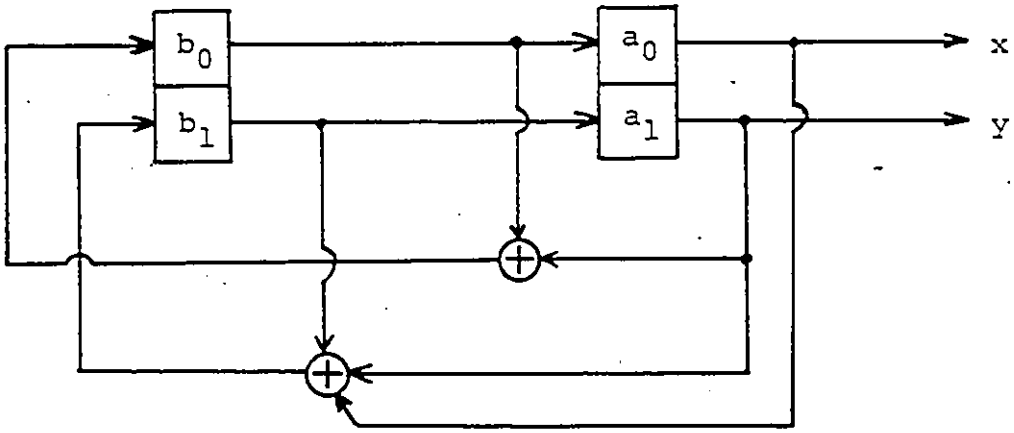


Fig. 4.1(b): The implementation of Fig. 4.1(a) by using binary shift registers.

Example 4.1.2:

Consider now m=6 and q=4, producing $n=2^6-1=63$ and $GF(4)=\{0,1,\gamma,\gamma^2\}$ with logic provided by $\gamma^2=\gamma+1$. Again from Table 3.2

$$h(x)=x^3+x^2+\gamma^2x+\gamma \ , \qquad (4.1.16)$$

is assigned as the parity check polynomial. The coset numbers corresponding to n=63 and q=4 are listed as follow:

1,4,6	15,60,51
2,8,32	21
3,12,48	22,25,37
5,20,17	23,29,53
6,24,33	26,41,38
7,28,49	27,45,54
9,36,18	30,57,39
10,40,34	31,61,55
11,44,50	42
13,52,19	43,46,58
14,56,35	47,62,59 .

(4.1.17)

Let the 63-bit code V generated by h(x) be $V=(v_0,v_1,v_2,\ldots,v_{62})$, the recursive formula can then be derived, from the fact that $V \cdot H=0$, as

$$v_i = v_{i-1} + \gamma^2 v_{i-2} + \gamma\, v_{i-3} \quad . \tag{4.1.18}$$

And the following equations can be written down:

$$\begin{aligned}
&\gamma v_0 + \gamma^2 v_1 + v_2 + v_3 = 0 \\
&\gamma v_1 + \gamma^2 v_2 + v_3 + v_1 = 0 \\
&\gamma v_2 + \gamma^2 v_3 + v_1 + v_5 = 0 \\
&\gamma v_3 + \gamma^2 v_1 + v_5 + v_6 = 0 \\
&\gamma v_1 + \gamma^2 v_5 + v_6 + v_7 = 0 \\
&\gamma v_5 + \gamma^2 v_6 + v_7 + v_2 = 0 \\
&\quad\vdots
\end{aligned} \tag{4.1.19}$$

As can be observed, if v_0 is fixed, the first six equations of (4.1.19) have only six variables namely v_1, v_2, v_3, v_5, v_6 and v_7, hence the following can be obtaned:

$$\begin{aligned}
v_1 &= v_0 \\
v_2 &= v_0 \\
v_3 &= 0 \\
v_5 &= \gamma^2 v_0 \\
v_6 &= 0 \\
v_7 &= 0 \quad .
\end{aligned} \tag{4.1.20}$$

From (4.1.18), at least three stages of initial conditions need to be known in order to generate the code sequences.

Since all v_i's can be expressed in terms of v_0, abitrary values to v_0, can be assigned thus generating the corresponding sequences. Because there are only three nonzero elements from GF(4), $v_0=1$, γ or γ^2, three c-m-sequences can be obtained according to (4.1.18) and (4.1.20) as

V_1 : 1 1 1 0 1 γ^2 0 0 1 1 γ γ^2 0 1 0 γ^2 1 γ^2 1 1 γ^2
γ γ γ 0 γ 1 0 0 γ γ γ^2 1 0 γ 0 1 γ 1 γ γ 1
γ^2 γ^2 γ^2 0 γ^2 γ 0 0 γ^2 γ^2 1 γ 0 γ^2 0 γ γ^2 γ γ^2 γ^2 γ

V_2 : γ γ γ 0 γ 1 0 0 γ γ γ^2 1 0 γ 0 1 γ 1 γ γ 1
γ^2 γ^2 γ^2 0 γ^2 γ 0 0 γ^2 γ^2 1 γ 0 γ^2 0 γ γ^2 γ γ^2 γ^2 γ
1 1 1 0 1 γ^2 0 0 1 1 γ γ^2 0 1 0 γ^2 1 γ^2 1 1 γ^2

V_3 : γ^2 γ^2 γ^2 0 γ^2 γ 0 0 γ^2 γ^2 1 γ 0 γ^2 0 γ γ^2 γ γ^2 γ^2 γ
1 1 1 0 1 γ^2 0 0 1 1 γ γ^2 0 1 0 γ^2 1 γ^2 1 1 γ^2
γ γ γ 0 γ 1 0 0 γ γ γ^2 1 0 γ 0 1 γ 1 γ γ 1

Let these c-m-sequences be designated $V_1(X)$, $V_2(X)$ and $V_3(X)$, corresponding to initial conditions of $v_0=1$, γ , γ^2 respectively. It can be observed that

$$V_2(X) = X^{42}\, V_1(X) \bmod (X^{63}+1)$$

and (4.1.21)

$$V_3(X) = X^{21}\, V_1(X) \bmod (X^{63}+1)$$

And all three sequences repeat themselves after 63 bits. Notice that the relation $v_i = v_{4i}$ does hold for all three sequences. Therefore, by using equation (4.1.16) and assigning three initial conditions to v_0, the three c-m-sequences over GF(4) can be generated.

According to (4.1.18), a 3-stage shift register device is needed in order to construct these c-m-sequences, with respect to three different initial conditions which are

$$(v_2,v_1,v_0) = \begin{cases} (1,1,1) \\ (\gamma,\gamma,\gamma) \\ (\gamma^2,\gamma^2,\gamma^2) \end{cases} . \qquad (4.1.22)$$

Figure 4.2(a) shows such a shift register device with feedback function determined by (4.1.16). Again, this device can be implemented on binary fashions by using the following adoptions:

$$\begin{aligned} v_0 &= a_0 + a_1\gamma \\ v_1 &= b_0 + b_1\gamma \\ v_2 &= c_0 + c_1\gamma , \end{aligned} \qquad (4.1.23)$$

where all coefficients are either 0 or 1. From equations (4.1.18) and (4.1.23), the following is obtained,

$$\begin{aligned} v_3 &= v_2+\gamma^2 v_1+\gamma v_0 \\ &= (c_0+c_1\gamma)+\gamma^2(b_0+b_1\gamma)+\gamma(a_0+a_1\gamma) \\ &= c_0+c_1\gamma+b_0\gamma^2+b_1+a_0\gamma+a_1\gamma^2 \\ &= (a_1+b_0+b_1+c_0) + (a_0+a_1+b_0+c_1)\gamma . \end{aligned} \qquad (4.1.24)$$

Figure 4.2(b) gives the implementation of Figure 4.2(a) by binary shift registers. The binary output (x,y) represents a symbol in GF(4) and thus the following:

$$v_i = x+y\gamma \quad , \quad \text{for} \quad 0 \le i \le n \ . \tag{4.1.25}$$

The 3 initial conditions to be loaded on the shift registers to construct the c-m-sequences thus become

$$(v_2,v_1,v_0) = ((c_0,c_1),(b_0,b_1),(a_0,a_1)) = \begin{cases} ((1.0),(1,0),(1,0)) \\ ((0,1),(0,1),(0,1)) \\ ((1,1),(1,1),(1,1)) \ . \end{cases} \tag{4.1.26}$$

Surely, the above consideration of binary implementations can be applied to any other Galois fields $GF(q=2^s)$. For example, if q=8, a 3-tuple over GF(2) is needed to represent a symbol in GF(8); if q=16, a 4-tuple over GF(2) is needed to represent a symbol in GF(16); and so forth.

From the above examples, the construction of c-m-sequences can be done methodically. At first one needs to express every v_i's in terms of v_0 (if $v_0 \neq 0$), or in terms of v_1 (if $v_0=0$). Then by choosing different nonzero initial conditions for v_0 (or v_1), q-1 c-m-sequences can be constructed by the recursive formula defined by the parity check polynomial h(x). This is because there are only q-1

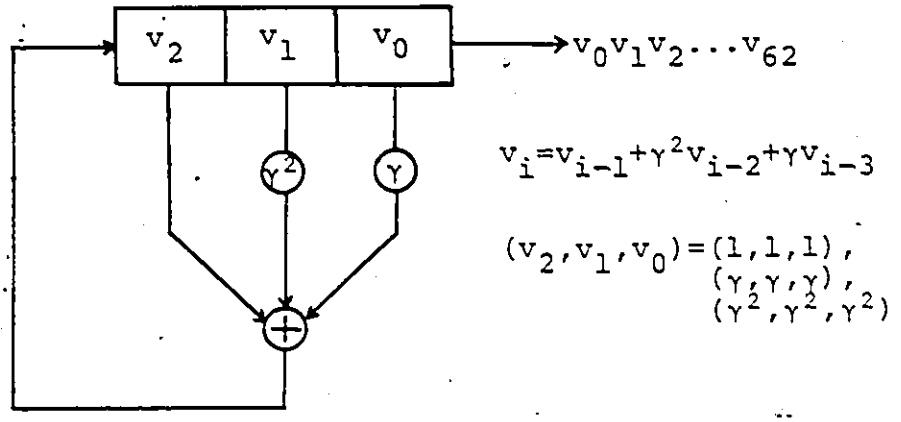


Fig. 4.2(a): A 3-stage shift register device for constructing c-m-sequences over GF(4) with $h(x)=x^3+x^2+\gamma^2x+\gamma$, where $\gamma^2=\gamma+1$.

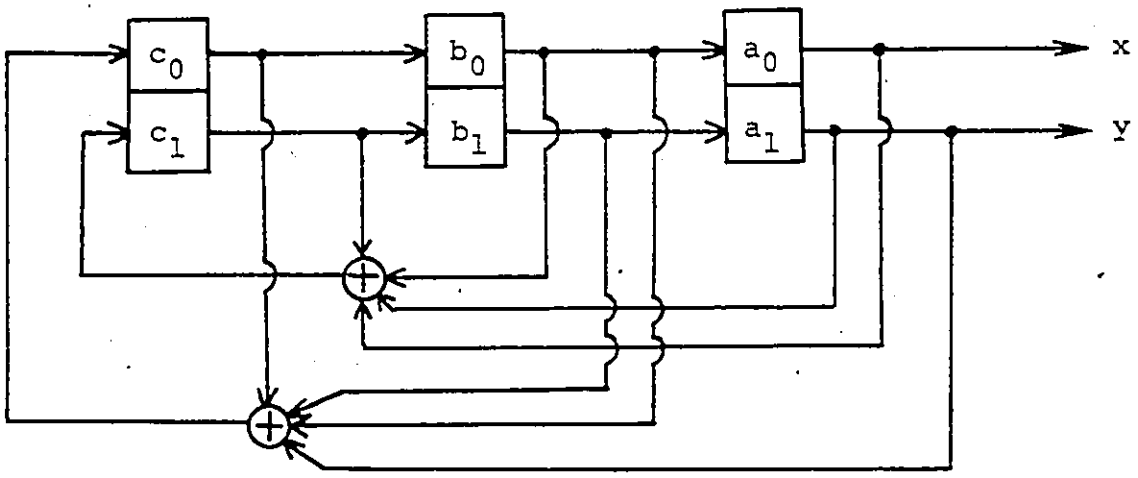


Fig. 4.2(b): The binary shift register implementation of Fig. 4.2(a).

nonzero elements which can be assigned to v_0 (or v_1) from GF(q). The number of stages that a shift register has in order to generate such sequences equals to the degree of h(x).

Table 4.1 shows some results for generating c-m-sequences over different Galois fields, according to the specified parity check polynomial h(x) which is chosen as one of the factors from Table 3.3. Also included in Table 4.1 are the feedback shift register circuits and the corresponding initial conditions in order to construct the c-m-sequences.

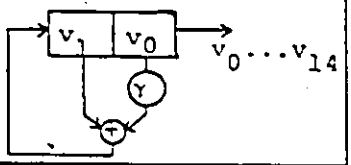
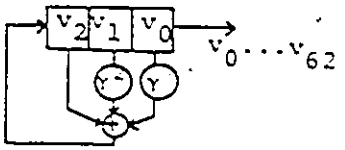
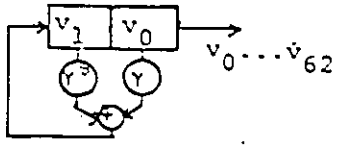
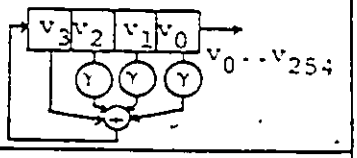
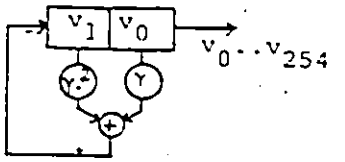
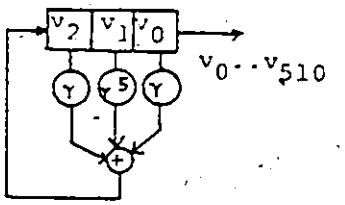
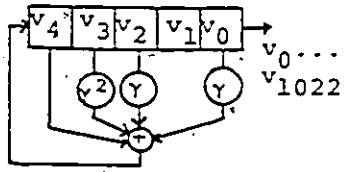
$n=2^m-1$	$q=2^s$	$h(x)$ over $GF(q)$	logic for $GF(q)$	initial cond'n for generating char. m-seqs.	corresponding shift register circuit
15	4	$x^2+x+\gamma$	$\gamma^2=\gamma+1$	$(v_1,v_0)=(1,0),(\gamma,0),(\gamma^2,0)$	
63	4	$x^3+x^2+\gamma^2x+\gamma^2$	$\gamma^2=\gamma+1$	$(v_2,v_1,v_0)=(1,1,1),(\gamma,\gamma,\gamma),(\gamma^2,\gamma^2,\gamma^2)$	
	8	$x^2+\gamma^3x+\gamma$	$\gamma^3=\gamma^2+1$	$(v_1,v_0)=(1,0),(\gamma,0),(\gamma^2,0),(\gamma^3,0),(\gamma^4,0),(\gamma^5,0),(\gamma^6,0)$	
255	4	$x^4+x^3+\gamma x^2+\gamma x+\gamma$	$\gamma^2=\gamma+1$	$(v_3,v_2,v_1,v_0)=(0,1,1,1).(0,\gamma,\gamma,\gamma),(0,\gamma^2,\gamma^2,\gamma^2)$	
	16	$x^2+\gamma^2x+\gamma$	$\gamma^4=\gamma+1$	$(v_1,v_0)=(1,0),(\gamma,0)$ $(\gamma^2,0),(\gamma^3,0)\ldots(\gamma^{14},0)$	
511	8	$x^3+\gamma x^2+\gamma^5x+\gamma$	$\gamma^3=\gamma+1$	$(v_2,v_1,v_0)=(\gamma^2,\gamma,1),(\gamma^3,\gamma^2,\gamma),(\gamma^4,\gamma^3,\gamma^2),\ldots,(\gamma^{11},\gamma^6)$	
1023	4	$x^5+x^4+\gamma^2x^3+\gamma x^2+\gamma$	$\gamma^2=\gamma+1$	$(v_4,v_3,v_2,v_1,v_0)=(1,0,1,1,1),(\gamma,0,\gamma,\gamma,\gamma),(\gamma^2,0,\gamma^2,\gamma^2,\gamma^2)$	

Table 4.1: Some examples for generation of characteristic m-sequences over GF(q) by h(x) as specified

4.2 THE SPECIAL CASE OF GF(q^2)

Consider now a parity check polynomial h(x) with degree 2. The shift register device used to generate the c-m-sequences has 2 stages. The following theorem deals with such a special case corresponding to a 2nd degree parity check polynomial.

Theorem II: With reference to

$$GF(q)= \{0,1,\gamma,\gamma^2,\ldots,\gamma^{\nu-1}\} \qquad (4.2.1)$$

where $\nu=q-1=2^s-1$ according to the logic provided by

$$h(x)= x^2+x+\gamma \qquad (4.2.2)$$

over GF(q) with α as a root, where α is the primitive element of GF(q^2). The sequence $v_0v_1v_2\ldots v_{n-1}$, where $n=q^2-1$, generated by h(x) over GF(q) is an m-sequence iff $Tr(\gamma)=1$. Furthermore, the m-sequence is characteristic iff $v_0=0$.

Proof:

By definition, if a primitive parity check polynomial h(x) over GF(q) which has degree k and exponent n, where

$n=q^k-1$, is used to generate a sequence $V=(v_0, v_1, v_2, \ldots, v_{n-1})$ over GF(q), V is an m-sequence. Since α is a root of $h(x)=x^2+x+\gamma$, where α is the primitive element of $GF(q^2)$, then

$$GF(q^2) = \{0,1,\alpha,\alpha^2, \ldots ,\alpha^{n-1}\} , \qquad (4.2.3)$$

V is an m-sequence if

$$h(\alpha) = 0 \quad \Rightarrow \qquad (4.2.4)$$
$$\alpha^2 = \gamma + \alpha$$

is indeed the logic to generate $GF(q^2)$. Prove first that $\alpha^j=1$ for j=0 or j=n, but not any other integer which is less than n and greater than 0. Since the 0th power of any number is 1, $\alpha^0=1$ is obvious. What is left now is to prove $\alpha^n=1$ for $n=q^2-1$. It has already been shown in equation (3.3.9) that if $h(x)=x^2+x+\gamma$ with α as a root, $\gamma=\alpha^{2^s+1}$ iff $Tr(\gamma)=1$. That is

$$\gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{s-1}} = 1 . \qquad (4.2.5)$$

By taking squares of (4.2.5), the following are obtained:

$$\begin{aligned}
&\gamma^2+\gamma^4+\gamma^8+\ldots+\gamma^{2^s} = 1 ,\\
&\gamma^4+\gamma^8+\gamma^{16}+\ldots+\gamma^{2^{s+1}} = 1 ,\\
&\qquad\vdots\\
&\gamma^{2^s}+\gamma^{2^{s+1}}+\gamma^{2^{s+2}}+\ldots+\gamma^{2^{2s-1}} = 1 .
\end{aligned} \qquad (4.2.6)$$

By squaring (3.3.7) repeatedly,

$$\alpha^{2^s} = \gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{s-1}}+\alpha$$

$$\alpha^{2^{s+1}} = \gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^s}+\alpha$$

$$\alpha^{2^{s+2}} = \gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{s+1}}+\alpha$$

$$\vdots$$

$$\alpha^{2^{2s}} = \gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{2s-1}}+\alpha \quad . \qquad (4.2.7)$$

Breaking the equation (4.2.7) into

$$\alpha^{2^{2s}} = (\gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{s-1}}) + (\gamma^{2^s}+\gamma^{2^{s+1}}+\ldots+\gamma^{2^{2s-1}})+\alpha \quad . \qquad (4.2.8)$$

With (4.2.5) and (4.2.6), the equation (4.2.8) becomes

$$\alpha^{2^{2s}} = 1+1+\alpha=\alpha$$

$$\alpha^{2^{2s}} = \alpha q^2 = \alpha$$

$$\alpha^{n+1} = \alpha \quad .$$

$$\alpha^n = 1 \quad . \qquad (4.2.9)$$

This proves that $\alpha^n=1$, but suppose there is another integer j', which is less than n and greater than 0, such that $\alpha^{j'}=1$. And suppose there exists an integer j, where

$$\alpha^j \neq \alpha \quad \text{for} \quad 1<j<n+1 \quad . \qquad (4.2.10)$$

It is known that $\alpha^2=\gamma+\alpha$ and since (4.2.10) assumes $\alpha^j=\alpha$, then

$$(\alpha^j)^2 = \alpha^{2j} = \alpha^2 = \gamma + \alpha,$$

and $$\alpha^{4j} = \gamma+ \gamma^2+ \alpha$$

$$\alpha^{8j} = \gamma+ \gamma^2+ \gamma^4+ \alpha$$

$$\vdots$$

$$\alpha^{2^k j} = \gamma +\gamma^2 +\gamma^4 + \ldots+\gamma^{2^{k-1}} + \alpha \quad , \qquad (4.2.11)$$

for any k. By comparing (4.2.11) with (3.3.7), obtained earlier, it can be concluded that $\alpha^{2^k j} = \alpha^{2^k}$. That is $j=1$, which contradicts the assumption of equation (4.2.10). Therefore, it proves that $\alpha^0=1$ and $\alpha^n=1$ are the only two cases existing, meaning that $\alpha^2= \gamma +\alpha$ is the logic to generate GF(q^2). Hence the sequence $v_0v_1v_2\ldots v_{n-1}$ produced from the logic provided by $h(x)=x^2+x+\gamma$ is an m-sequence. The next thing to do is to prove that the m-sequence is characteristic iff $v_0=0$. From $V\cdot H=0$ it is known that

$$\gamma v_0 + v_1 + v_2 = 0 \quad , \qquad (4.2.12)$$

or equivalently

$$v_2 = v_1+\gamma v_0$$

$$v_3 = v_2+\gamma v_1 = (v_1+\gamma v_0)+\gamma v_1$$
$$= (\gamma+1)v_1+\gamma v_0$$

$$v_4 = v_3+\gamma v_2 = (\gamma+1)v_1+\gamma v_0+\gamma(v_1+\gamma v_0)$$
$$=(\gamma+\gamma^2)v_0+v_1$$

$$v_5 = v_4+\gamma v_3 = v_0+(\gamma^2+\gamma+1)v_1$$

$$\vdots$$

$$v_8 = v_7+\gamma v_6 = (\gamma+\gamma^2+\gamma^4)v_0+v_1$$

$$\vdots$$

$$v_{16} = v_{15}+\gamma v_{14} = (\gamma+\gamma^2+\gamma^4+\gamma^8)v_0+v_1$$

$$\vdots$$

$$v_{2^s} = v_{2^s-1} + \gamma v_{2^s-2}$$

$$= (\gamma+\gamma^2+\gamma^4+\ldots+\gamma^{2^{s-1}})v_0 + v_1$$

$$= Tr(\gamma)\ v_0 + v_1 \quad . \tag{4.2.13}$$

Since $Tr(\gamma)=1$,

$$v_{2^s} = v_0 + v_1 \quad . \tag{4.2.14}$$

The c-m-sequences have an unique property, which is $v_i=v_{qi}$ where $q=2^s$ for any i. When $i=1$,

$$v_1 = v_q = v_{2^s} \quad . \tag{4.2.15}$$

Compare (4.2.14) with (4.2.15), $v_{2^s}=v_1=v_1+v_0$, which implies that $v_0=0$. Therefore the m-sequence is characteristic iff $v_0=0$; which completes the proof for Theorem II. Q.E.D.

Now some examples will be given to show the validity for Theorem II.

Example 4.2.1:

Examine example 4.1.1, where $m=4$, $n=2^4-1=15$ and $q=4$, the parity check polynomial $h(x)=x^2+x+\gamma$ which is the case when the corresponding extension field is $GF(q^2)$. The logic provided for $h(x)$ is $\gamma^2=\gamma+1$. It is clear that $Tr(\gamma)=\gamma^2+\gamma=1$. As can be observed from (4.1.10), the three sequences generated by $h(x)$ were indeed a cyclic shift of each others, and these sequences are characteristic since $v_i=v_{4i}$, and $v_0=0$. Therefore, with reference to the polynomial $x^2+x+\gamma$ with α as a root, where α is the primitive element of the corresponding extension field $GF(q^2)$, the sequences with length $n=2^m-1=q^2-1$ constructed by $h(x)$ are m-sequences when $Tr(\gamma)=1$. Furthermore, these sequences are characteristic iff $v_0=0$. Theorem II thus holds.

Example 4.2.2:

In the case of $m=6$, $n=2^6-1=63$, and $q^2=2^6$ which implies $q=8$. The corresponding binary primitive irreducible polynomial is $f(x)=x^6+x^5+x^3+x^2+1$ as listed in Table 3.4. The ground field with γ as the primitive element and the

corresponding extension field with α as the primitive element can be expressed as

$$GF(8)=\{0,1,\gamma,\gamma^2,\ldots,\gamma^6\} \ , \quad \gamma^7 = 1 \ ,$$
$$GF(8^2)=\{0,1,\alpha,\alpha^2,\ldots,\alpha^{62}\} \ , \quad \alpha^{63} = 1 \ , \tag{4.2.16}$$

with the logic provided for GF(8) as $\gamma^3 = \gamma^2+1$. Hence the following:

$$\gamma^3 = \gamma^2 + 1 \ ,$$
$$\gamma^4 = \gamma^2+ \gamma+ 1 \ ,$$
$$\gamma^5 = \gamma+ 1 \ ,$$
$$\gamma^6 = \gamma^2+ \gamma \ ,$$
$$\gamma^7 = 1 \ . \tag{4.2.17}$$

And $Tr(\gamma)=\gamma+\gamma^2+\gamma^4=\gamma+\gamma^2+\gamma^2+\gamma+1=1$. Let $V=(v_0,v_1,v_2,\ldots,v_{62})$ represents the code generated by the parity check polynomial $h(x)=x^3+x+\gamma$. The coset numbers for q=8 and n=63 are

1,8	14,49	30,51
2,16	15,57	31,59
3,24	18	36
4,32	19,26	37,44
5,40	20,34	38,52
6,48	21,42	39,60
7,56	22,50	45

9	23,58	46,53
10,17	27	47,61
11,25	28,35	54
12,33	29,43	55,62
13,41		

Since V·H=0, the following are obtained,

$$\begin{aligned}
\gamma v_0 + v_1 + v_2 &= 0 \\
\gamma v_1 + v_2 + v_3 &= 0 \\
\gamma v_2 + v_3 + v_4 &= 0 \\
\gamma v_3 + v_4 + v_5 &= 0 \\
\gamma v_4 + v_5 + v_6 &= 0 \\
\gamma v_5 + v_6 + v_7 &= 0 \\
\gamma v_6 + v_7 + v_1 &= 0 \\
&\vdots
\end{aligned} \tag{4.2.18}$$

It can be observed from (4.2.18) that if v_0 is fixed, there are only 7 variables in the top 7 equations, namely v_1, v_2, v_3, v_4, v_5, v_6 and v_7. Hence these equations can be solved and the following solutions are obtained,

$$\begin{aligned}
v_0 &= 0 \\
v_2 &= v_1 \\
v_3 &= \gamma^5 v_1
\end{aligned}$$

$$v_4 = v_1$$
$$v_5 = \gamma^4 v_1$$
$$v_6 = \gamma^3 v_1$$
$$v_7 = \gamma^6 v_1 \qquad (4.2.19)$$

The recursive equation according to h(x) is

$$v_i = v_{i-1} + \gamma v_{i-2} \qquad (4.2.20)$$

By using (4.2.19) and (4.2.20), all v_i's can be expressed in terms of v_1. Therefore the code V generated by h(x) contains only 7 sequences, it is because there are only 7 nonzero elements from GF(8) which can be assigned to v_1 in order to generate these sequences. These are designated V_1, V_2, V_3, V_4, V_5, V_6 and V_7 as the sequences generated corresponding to v_1=1, γ, γ^2, γ^3, γ^4, γ^5 and γ^6 respectively. These sequences can then be obtained by applying (4.2.19) and (4.2.20) as follows:

V_1: $0\ 1\ 1\ \gamma^5\ 1\ \gamma^4\ \gamma^3\gamma^6\ 1\ 0\ \gamma\ \gamma\ \gamma^6\ \gamma\ \gamma^5\gamma^4\ 1\ \gamma\ 0\ \gamma^2\ \gamma^2\ 1\ \gamma^2\gamma^6\ \gamma^5\ \gamma\ \gamma^2$
$0\ \gamma^3\ \gamma^3\ \gamma\ \gamma^3\ 1\ \gamma^6\gamma^2\ \gamma^3\ 0\ \gamma^4\gamma^4\ \gamma^2\ \gamma^4\gamma\ 1\ \gamma^3\gamma^4\ 0\ \gamma^5\ \gamma^5\ \gamma^3\gamma^5\gamma^2\ \gamma\ \gamma^4\ \gamma^5$
$0\ \gamma^6\ \gamma^6\gamma^4\ \gamma^6\gamma^3\ \gamma^2\gamma^5\ \gamma^6$

V_2: $0\ \gamma\ \gamma\ \gamma^6\ \gamma\ \gamma^5\ \gamma^4\ 1\ \gamma\ 0\ \gamma^2\gamma^2\ 1\ \gamma^2\gamma^6\gamma^5\ \gamma\ \gamma^2\ 0\ \gamma^3\ \gamma^3\ \gamma\ \gamma^3\ 1\ \gamma^6\gamma^2\ \gamma^3$
$0\ \gamma^4\ \gamma^4\gamma^2\ \gamma^4\ \gamma\ 1\ \gamma^3\ \gamma^4\ 0\ \gamma^5\gamma^5\ \gamma^3\ \gamma^5\gamma^2\gamma\ \gamma^4\gamma^5\ 0\ \gamma^6\ \gamma^6\ \gamma^4\gamma^6\gamma^3\ \gamma^2\gamma^5\ \gamma^6$
$0\ 1\ 1\ \gamma^5\ 1\ \gamma^4\ \gamma^3\gamma^6\ 1$

V_3: $0\ \gamma^2\ \gamma^2\ 1\ \gamma^2\gamma^6\ \gamma^5\ \gamma\ \gamma^2\ 0\ \gamma^3\gamma^3\ \gamma\ \gamma^3 1\ \gamma^6\ \gamma^2\gamma^3\ 0\ \gamma^4\ \gamma^4\ \gamma^2\gamma^4\gamma\ 1\ \gamma^3\ \gamma^4$
$0\ \gamma^5\ \gamma^5\gamma^3\ \gamma^5\gamma^2\ \gamma\ \gamma^4\ \gamma^5\ 0\ \gamma^6\gamma^6\ \gamma^4\ \gamma^6\gamma^3\gamma^2\ \gamma^5\gamma^6\ 0\ 1\ 1\ \gamma^5 1\ \gamma^4\ \gamma^3\gamma^6\ 1$
$0\ \gamma\ \gamma\ \gamma^6\ \gamma\ \gamma^5\ \gamma^4\ 1\ \gamma$

V_4: $0\ \gamma^3\ \gamma^3\gamma\ \ \gamma^3\ 1\ \gamma^6\gamma^2\ \gamma^3\ 0\ \gamma^4\gamma^4\ \gamma^2\ \gamma^4\gamma\ 1\ \ \gamma^3\gamma^4\ 0\ \gamma^5\ \gamma^5\ \gamma^3\gamma^5\gamma^2\ \gamma\ \gamma^4\ \gamma^5$

$0\ \gamma^6\ \gamma^6\gamma^4\ \gamma^6\gamma^3\ \gamma^2\gamma^5\ \gamma^6\ 0\ 1\ 1\ \ \gamma^5\ 1\ \gamma^4\gamma^3\ \gamma^6\ 1\ 0\ \gamma\ \ \gamma\ \gamma^6\gamma\ \gamma^5\ \gamma^4\ 1\ \gamma$

$0\ \gamma^2\ \gamma^2\ 1\ \gamma^2\gamma^6\ \gamma^5\gamma\ \ \gamma^2$

V_5: $0\ \gamma^4\ \gamma^4\gamma^2\ \gamma^4\gamma\ \ 1\ \gamma^3\ \gamma^4\ 0\ \gamma^5\gamma^5\ \gamma^3\ \gamma^5\gamma^2\gamma\ \ \gamma^4\gamma^5\ 0\ \gamma^6\ \gamma^6\ \gamma^4\gamma^6\gamma^3\ \gamma^2\gamma^5\ \gamma^6$

$0\ 1\ \ 1\ \gamma^5\ 1\ \gamma^4\ \gamma^3\gamma^6\ 1\ \ 0\ \gamma\ \gamma\ \ \gamma^6\ \gamma\ \gamma^5\gamma^4\ 1\ \gamma\ \ 0\ \gamma^2\ \gamma^2\ 1\ \gamma^2\gamma^6\ \gamma^5\gamma\ \ \gamma^2$

$0\ \gamma^3\ \gamma^3\gamma\ \ \gamma^3\ 1\ \gamma^6\gamma^2\ \gamma^3$

V_6: $0\ \gamma^5\ \gamma^5\gamma^3\ \gamma^5\gamma^2\ \gamma\ \gamma^4\ \gamma^5\ 0\ \gamma^6\gamma^6\ \gamma^4\ \gamma^6\gamma^3\gamma^2\ \gamma^5\gamma^6\ 0\ 1\ \ 1\ \ \gamma^5\ \ \gamma^4\ \gamma^3\gamma^6\ 1$

$0\ \gamma\ \ \gamma\ \gamma^6\ \gamma\ \gamma^5\ \gamma^4\ 1\ \gamma\ \ 0\ \gamma^2\gamma^2\ \ 1\ \gamma^2\gamma^6\gamma^5\ \gamma\ \gamma^2\ 0\ \gamma^3\ \gamma^3\ \gamma\ \gamma^3\ 1\ \gamma^6\gamma^2\ \gamma^3$

$0\ \gamma^4\ \gamma^4\gamma^2\ \gamma^4\gamma\ \ 1\ \gamma^3\ \gamma^4$

V_7: $0\ \gamma^6\ \gamma^6\gamma^4\ \gamma^6\gamma^3\ \gamma^2\gamma^5\ \gamma^6\ 0\ 1\ \ 1\ \gamma^5\ 1\ \gamma^4\gamma^3\ \gamma^6\ 1\ 0\ \gamma\ \ \gamma\ \ \gamma^6\gamma\ \gamma^5\ \gamma^4\ 1\ \gamma$

$0\ \gamma^2\ \gamma^2\ 1\ \gamma^2\gamma^6\ \gamma^5\gamma\ \ \gamma^2\ 0\ \gamma^3\gamma^3\ \gamma\ \ \gamma^3 1\ \gamma^6\ \gamma^2\gamma^3\ 0\ \gamma^4\ \gamma^4\ \gamma^2\gamma^4\gamma\ \ 1\ \gamma^3\ \gamma^4$

$0\ \gamma^5\ \gamma^5\gamma^3\ \gamma^5\gamma^2\ \gamma\ \gamma^4\ \gamma^5$

(4.2.21)

These sequences repeat themselves after 63 bits. As observed,

$$V_2(X) = X^{54}\ V_1(X) \bmod (X^{63}+1)$$

$$V_3(X) = X^{45}\ V_1(X) \bmod (X^{63}+1)$$

$$V_4(X) = X^{36}\ V_1(X) \bmod (X^{63}+1)$$

$$V_5(X) = X^{27}\ V_1(X) \bmod (X^{63}+1)$$

$$V_6(X) = X^{18}\, V_1(X) \bmod (X^{63}+1)$$
$$V_7(X) = X^{9}\, V_1(X) \bmod (X^{63}+1) \quad . \qquad (4.2.22)$$

Each of these sequences is a cyclic shift of others, and with maximum-length being 63, they can be generated by primitive feedback shift register of 2 stages, hence each sequence is an m-sequence over GF(q). Since h(x) is a polynomial with degree 2, there are two stages in the shift register device in which the feedback connection is determined by h(x). The seven initial conditions which the shift register device used to generate the sequences are: $(v_1,v_0)=(1,0)$, $(\gamma,0)$, $(\gamma^2,0)$, $(\gamma^3,0)$, $(\gamma^4,0)$, $(\gamma^5,0)$, and $(\gamma^6,0)$. Figure 4.3 shows such a 2-stage shift register device. It is obvious that $v_i=v_{8i}$ in all seven sequences of (4.2.21), hence these sequences are c-m-sequences over GF(8). It has been shown before that $Tr(\gamma)=1$, and as observed from (4.2.21), $v_0=0$ in this case; hence Theorem II holds.

Since $h(x)=x^2+x+\gamma$ can be used to construct c-m-sequences over different Galois fields provided by different logics, the shift register circuits used to generate such sequences are always the same as shown either in Figure 4.1(a) or Figure 4.3. The only difference is the number of initial

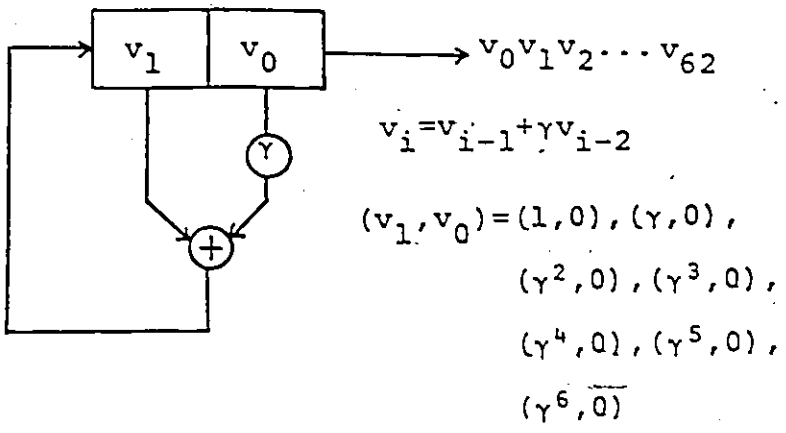


Fig. 4.3: A 2-stage shift register device for constructing c-m-sequences over GF(8) by $h(x)=x^2+x+\gamma$, where $\gamma^3=\gamma^2+1$.

conditions to be loaded in the shift registers for different Galois fields. This number is q-1 and the corresponding initial conditions are $(v_1,v_0)=$. $(1,0)$, $(\gamma,0)$, $(\gamma^2,0)$, ..., $(\gamma^{q-2},0)$. Of course it should be noticed that there is different logic provided for different Galois fields, the design of binary shift register implemenation is based on individual logic provided.

A similar approach is used to find the initial conditions to generate the c-m-sequences over different GF(q)'s by some logics provided as shown by Table 3.4. It is found that Theorem II holds in all our examples. Hence, Theorem II is satisfied by all our examples.

In general, once the parity check polynomial h(x) has been determined with some degree, it can be used recursively according to some logic and thus generate the q-nary c-m-sequences by specifying q. It can be done by computer programming, all that is needed is to represent the elements of GF(q) such as γ, γ^2, etc., into some numbers, and then write the program according to the specific logic used. For example, a computer program was written to compute the c-m-sequences over GF(4) by different parity check polynomials, and the logic provided for GF(4) is $\gamma^2=\gamma+1$. The output prints all q-1=3 c-m-sequences for different h(x). In the program, '1' represents γ, '2' represents γ^2, '3' represents $1(=\gamma^3)$ and '0' represents 0. The output sequences match the results obtained earlier in the examples. In the case of other Galois fields being used, one only needs to modify the program slightly according to the different logic provided for different Galois fields. Hence, with the aid of computer programs, even for larger length of sequences, one can still compute the corresponding c-m-sequences and thus can observe the behaviors of these sequences with detail.

Chapter V

GENERAL ASPECTS OF q-NARY C-M-SEQUENCES

As shown above in Chapter IV, the q-nary c-m-sequences can be computed for any length n where $n=2^m-1$ with respect to different GF(q)'s where $q=2^s$. Furthermore, $n=q^k-1$ for some positive integer k where k is the degree of the corresponding primitive parity check polynomial. It is obvious that the set of all q-nary c-m-sequences with length n is a subset of the set of all q-nary m-sequences with the same length n, with respect to the same logic provided for GF(q). Therefore, c-m-sequences have all, plus some additional properties, of m-sequences. These properties of general m-sequences will be discussed, and also the additional, special behaviors of c-m-sequences, where the error-correcting capability and majority logic decodable capability of c-m-sequences are studied.

5.1 PROPERTIES OF C-M-SEQUENCES OVER GF(q)

Some observed behaviors of q-nary m-sequences are listed below. Since the set of c-m-sequences is a subset of the set of m-sequences, what is true for m-sequences is also true for c-m-sequences. Also included are the special properties of the c-m-sequences that the m-sequences do not have.

P1: The q-nary m-sequences (and the c-m-sequences) have length n, where $n=2^m-1=q^k-1$. It is the maximum period possible for a linear feedback shift register of k stages since h(x) is chosen to be a primitive polynomial. In the binary case where q=2 and k=m, the length of m-sequences thus become 2^m-1.

P2: Each of the code sequences is a cyclic shift of others. In the q-nary m-sequences, the total number of sequences generated by h(x) which has degree k, is q^k-1. It is because, given any code sequence, all q^k-1 cyclic shifts of it must be code sequences too. But there are only q-1 c-m-sequences over GF(q) that can be generated by h(x) because of the extra restriction $v_i=v_{qi}$. Since all v_i's can be expressed in terms of one variable, v_0 (if $v_0\neq 0$) or v_1 (if $v_0=0$), the number of nonzero elements from GF(q) which can be assigned to v_0 or v_1 is only q-1. Hence, there are only q-1 c-m-sequences over GF(q) that can be constructed by h(x).

P3: For m-sequences, each sequence is a cyclic shift of one bit of another sequence. But if the m-sequences are characteristic, each of the sequences is a cyclic shift

of n/(q-1) bits of another sequence. For example, let $V_1(X)$, $V_2(X)$, $V_3(X)$,... , $V_{q-1}(X)$ be designated as the c-m-sequences generated by using 1, γ , γ^2 , γ^3 , ..., γ^{q-2} as the initial values being assigned to v_0 or v_1 respectively, and let j=n/(q-1), then

$$V_2(X) = X^{j(q-2)} V_1(X) \bmod (X^n+1)$$

$$V_3(X) = X^{j(q-3)} V_1(X) \bmod (X^n+1)$$

$$V_4(X) = X^{j(q-4)} V_1(X) \bmod (X^n+1)$$

$$\vdots$$

$$V_{q-2}(X) = X^{2j} V_1(X) \bmod (X^n+1)$$

$$V_{q-1}(X) = X^{j} V_1(X) \bmod (X^n+1) \quad . \qquad (5.1.1)$$

P4: The balance property still holds for q-nary m-sequences. That is, the number of appearances of all nonzero elements are equal and this number is q^{k-1}. Therefore the number of appearances of zero is $(q^k-1)-(q-1)q^{k-1}=q^{k-1}-1$. In the binary case, it becomes the number of 1's and number of 0's are different from one.

P5: The Hamming weight of a q-nary m-sequence is the number of all nonzero elements in this sequence. That is

$$\text{Hamming weight}=(q-1)q^{k-1}=q^k-q^{k-1}. \qquad (5.1.2)$$

For q=2, the Hamming weight is simply 2^{k-1}. Let N(w) denote the number of sequences which have weight w, the weight distribution of q-nary m-sequences thus becomes,

$$\left.\begin{array}{l} N(0)=1 \\ N(q^k-q^{k-1})=q^k-1=n. \end{array}\right\} \qquad (5.1.3)$$

This is analogous to the binary case when q=2.

P6: For c-m-sequences with length q^k-1, if a nonzero sequence is compared to another nonzero sequence within a period, term by term, the only agreements are the 0's, since all nonzero elements are different from one sequence to another. While there are $q^{k-1}-1$ zeros in a c-m-sequence, the number of disagreements(D) differs from the number of agreements(A) by $A-D=q^{k-1}-1-(q-1)q^{k-1}=(2-q)q^{k-1}-1$. This is also found true for the general m-sequences over GF(q). Therefore, the auto-correlation function of q-nary m-sequences of length $n=2^m-1=q^k-1$ is given by $\theta(j)=(A-D)/n$, thus the following can be written,

$$\theta(j) = \begin{cases} 1, & j=0, \\ \dfrac{(2-q)q^{k-1}-1}{q^{k-1}-1}, & 1 \leq j < n. \end{cases} \tag{5.1.4}$$

In the binary case when $q=2$ and $k=m$, equation (5.1.4) is reduced to equation (2.3.1), meaning that the number of disagreements and the number of agreements differs by one. It is also observed that the minimum distance between any two nonzero code sequences is just the Hamming weight of any one sequence,

$$d_{min}=(q-1)q^{k-1}. \tag{5.1.5}$$

P7: If an m-sequence over GF(q) is denoted by $v_0v_1v_2\cdots v_{n-1}$, and let $j=n/(q-1)$. It is very interesting to find that the sequence is divided into q-1 segments. Suppose these segments are denoted as

$$\begin{aligned} S_1 &= (v_0v_1v_2\cdots v_{j-1}) \\ S_2 &= (v_jv_{j+1}v_{j+2}\cdots v_{2j-1}) \\ S_3 &= (v_{2j}v_{2j+1}v_{2j+2}\cdots v_{3j-1}) \\ &\vdots \\ S_{q-1} &= (v_{(q-2)j}v_{(q-2)j+1}v_{(q-2)j+2}\cdots v_{n-1}). \end{aligned} \tag{5.1.6}$$

It is observed that

$$S_2 = \gamma S_1 \bmod \gamma^{q-1}$$

$$S_3 = \gamma^2 S_1 \bmod \gamma^{q-1}$$

$$S_4 = \gamma^3 S_1 \bmod \gamma^{q-1}$$

$$\vdots$$

$$S_{q-1} = \gamma^{q-2} S_1 \bmod \gamma^{q-1} \quad . \qquad (5.1.7)$$

This is an important property since one can indeed generate the whole sequence by computing only the first segment. Then by simply multiplication the rests of these segments can be obtained. Once a complete sequence has been obtained, apply equation (5.1.1) to yield the total q-1 code sequences. Thus the computation is much simplfied. Furthermore, by investigating one of these segments of the m-sequences, the properties of m-sequences can be observed within limited amount of time.

These are generalized properties of q-nary m-sequences as well as c-m-sequences. There are also other important factors which make these sequences become vital roles in many communication systems, such as the error-correcting

capability and majority logic decodable capability, which will be discussed in the next section.

5.2 ERROR-CORRECTING AND MAJORITY LOGIC DECODABLE CAPABILITIES OF C-M-SEQUENCES

Let D denote the minimum distance of c-m-sequence codes in q-nary field. From property P6, discussed in last section, D is equal to the Hamming weight of this code, hence,

$$D=(q-1)q^{k-1} . \tag{5.2.1}$$

Recall that the error-correcting capability of cyclic codes, t, is given by

$$t = [(D-1)/2] , \tag{5.2.2}$$

where [] denotes the largest integer which is less than the quantity inside the bracket []. Hence, by substituting (5.2.1) into (5.2.2),

$$t=[(q^k -q^{k-1} -1)/2] . \tag{5.2.3}$$

Therefore given the q-nary c-m-sequences with length q^k-1, this code can correct at most t errors.

The majority logic decoding is an effective decoding scheme, especially for certain classes of cyclic codes such

as m-sequence codes. There has been much, successful research done on majority logic decodable cyclic codes and the shift register circuits provided for such decoding schemes are given in many books such as Peterson & Weldon[14] and S. Lin[10].

The concepts of orthogonal parity check sums are central to majority logic decoding. Let e_1, e_2,..., e_i denote the error digits and let s_1, s_2, ..., s_j denote the check sums of various error digits. If a particular error digit e_1, weighted by a coefficient a from GF(q), is involved in every sum in the set and no other error digit is checked by more than one sum, then the sums are said to be orthogonal on e_1. More generally, if every sum checks e_1, e_2, ...,e_i with coefficients a, b, ... , j and no other error digits appear in more than one sum, then the check sums are orthogonal on the sum $ae_1+be_2+ce_3+...+je_i$.

It is well known that if one is able to construct a set of D-1 check sums orthogonal on any digit in a cyclic code, then the code has minimum distance at least D. Consequently, orthogonal check sums provide a lower bound on minimum distance. A cyclic code with minimum distance D is said to be completely orthogonalizable in one step iff it is possible to form J=D-1 parity check sums orthogonal on every error digit. The decoding procedures for one-step majority logic decodable cyclic codes are very simple. Figure 5.1

gives a one-step majority logic decoder for a cyclic (n,k) code, which is given by Perterson & Weldon [14, section 10.1], where the decoding procedures are explained step by

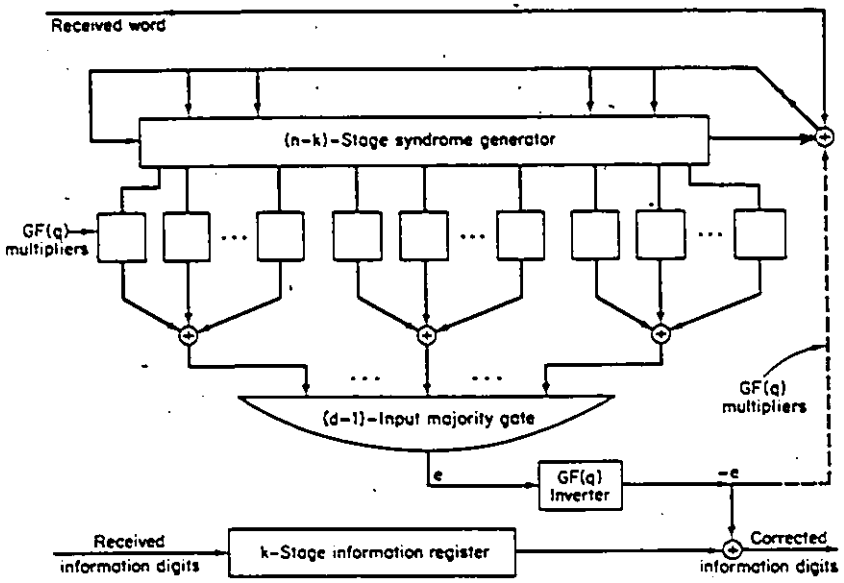


Fig. 5.1: A one-step majority logic decoder for a cyclic (n,k) code.

step.

Even all binary m-sequences are known to be one-step majority logic decodable. In general, not all q-nary m-sequences are. Hence it leads to the study of c-m-sequences. The examples given below illustrates this result.

Example 5.2.1:

Let $m=4$, $n=2^4-1=15$, and let $h(x)=x^4+x+\gamma$ over GF(4) where $GF(4)=\{0,1,\gamma,\gamma^2\}$, $\gamma^2=\gamma+1$. The m-sequences generated by $h(x)$, with length 15, are denoted by $v_0v_1v_2\ldots v_{14}$. Since $V\cdot H=0$, the following can be written,

$$\begin{aligned}
\gamma v_0 + v_1 + v_2 &= 0 \\
\gamma v_1 + v_2 + v_3 &= 0 \\
\gamma v_2 + v_3 + v_4 &= 0 \\
&\vdots
\end{aligned} \tag{5.2.4}$$

From equation (5.1.7), the property of m-sequences, $v_6=\gamma v_1$ and $v_{11}=\gamma^2 v_1$, equivalently

$$\begin{aligned}
v_1 &= \gamma^2 v_6 \\
&= \gamma v_{11}
\end{aligned} \tag{5.2.5}$$

By arranging the top three equations of (5.2.4), $v_1=\gamma v_0+v_2=v_3+\gamma v_4$, and also $v_0=\gamma^2 v_5=\gamma v_{10}$, $v_2=\gamma^2 v_7=\gamma v_{12}$, $v_3=\gamma^2 v_8=\gamma v_{13}$, $v_4=\gamma^2 v_9=\gamma v_{14}$. Together with (5.2.5), the following check sums on v_1 are obtained,

$$\begin{aligned}
v_1 &= \gamma^2 v_6 \\
&= \gamma v_{11} \\
&= \gamma v_0 + v_2 \\
&= v_5 + \gamma^2 v_7 \\
&= \gamma^2 v_{10} + \gamma v_{12} \\
&= v_3 + \gamma v_4
\end{aligned}$$

$$v_1 = \gamma^2 v_8 + v_9$$

$$= \gamma v_{13} + \gamma^2 v_{14} \quad . \tag{5.2.6}$$

All 8 orthogonal check sums on v_1 have been formed since all $v_0, v_1, \ldots, v_{14}$ have been used, and each of the v_i's appeared only once. Now determine the minimum distance D of these m-sequences,

$$D = (q-1)q^{k-1} = 12 \quad . \tag{5.2.7}$$

Since all v_i's have been used to form 8 orthogonal check sums on v_1, it is obvious that one cannot obtain J=D-1=11 orthogonal check sums on v_1. Therefore it must be concluded that this q-nary m-sequences code is not one-step majority logic decodable.

It has been illustrated from example 5.2.1 that it is not possible to form J=D-1 orthogonal check sums on v_1 (or v_0 if $v_0 \neq 0$) for q-nary m-sequences. It leads to the fact that not all q-nary m-sequences are one-step majority logic decodable. However, by introducing the constraint $v_i = v_{qi}$, these q-nary m-sequences become characteristic and they are indeed one-step majority logic decodable. The following explains. For c-m-sequences generated by h(x) with degree k over GF(q) provided by some logic, denote such sequences as $v_0 v_1 v_2 \ldots v_{n-1}$, where $n=q^k-1$. Try to find the orthogonal

check sums on v_0 (if $v_0 \neq 0$) or v_1 (if $v_0 = 0$) for c-m-sequences. It has been shown earlier that one can indeed express all nonzero v_i's in terms of v_0 (or v_1). Since there are $(q-1)q^{k-1} = D$ nonzero elements in every c-m-sequence, there are D-1 expressions on v_0 (or v_1), and each of these expressions contains one nonzero element from the c-m-sequence. For the zero valued v_i's, simply add them to any one of these expressions because they don't have any effect on these check sums. Thus, one can use all nonzero elements and form D-1 orthogonal check sums on v_0 (or v_1) since each of these v_i's appears once and only once in the check sums. Consequently, this ensures that there are D-1 orthogonal check sums that can be formed on v_0 (or v_1) for c-m-sequences. Equivalently, c-m-sequences are one-step majority logic decodable. To illustrate this, the constraint $v_i = v_{4i}$ is added to example 5.2.1.

Example 5.2.2:

By adding $v_i = v_{4i}$, example 5.2.1 becomes the same as example 4.1.1. From (4.1.7) and (4.1.9), $v_0 = 0$, $v_1 = v_2 = \gamma v_3 = v_4 = \gamma^2 v_6 = \gamma^2 v_7 = v_8 = \gamma^2 v_9 = \gamma v_{11} = \gamma v_{12} = \gamma^2 v_{13} = \gamma v_{14}$, and also $v_5 = v_{10} = 0$. By adding zero valued v_i's to expressions of v_1's, the following orthogonal check sums on v_1 can be obtained,

$$
\begin{aligned}
v_1 &= v_2 + v_0 \\
&= \gamma v_3 + v_5 \\
&= v_4 + v_{10} \\
&= \gamma^2 v_6 \\
&= \gamma^2 v_7 \\
&= v_8 \\
&= \gamma^2 v_9 \\
&= \gamma v_{11} \\
&= \gamma v_{12} \\
&= \gamma^2 v_{13} \quad = \gamma v_{14}
\end{aligned}
\tag{5.2.8}
$$

Therefore one can form J=11=D-1 orthogonal check sums on v_1, which implies that the corresponding c-m-sequences are one-step majority logic decodable and can correct up to t=[(12-1)/2]=5 errors.

Chapter VI

CONCLUSIONS

Binary m-sequences have been used very extensively in many communication systems such as spread spectrum, radar ranging, cryptographic and multiple access systems. However, for sophisticated applications, the q-nary m-sequences were studied since they can reduce the bandwidth required to transmit such sequences and they can also be generated by simple linear feedback shift registers circuits. Furthermore, by adding a constraint to the q-nary m-sequences, the c-m-sequences over GF(q) were obtained, where $q=2^s$.

First, the basic finite field theory and concepts of minimal polynomials were introduced since they are essential for establishing the factorization methods of binary primitive polynomials over GF(q). As illustrated by some examples, such factorizations can be done methodically. All the factors obtained were irreducible over GF(q) and one of these primitive factors was used to construct the q-nary m-sequences. The linear feedback shift registers used for individual examples were included. With the computed initial conditions being loaded in such shift registers, c-m-sequences were constructed over GF(q) for different logics provided.

Two theorems were derived with respect to the case when the corresponding extension field was $GF(q^2)$. With a particular chosen parity check polynomial with degree 2, q-nary c-m sequences were then constructed by a 2-stage shift register with feedback connections determined by the corresponding primitive parity check polynomial h(x).

Finally, the properties of m-sequences as well as c-m-sequences over GF(q) were discussed. It was also shown that because of the special properties which the q-nary m-sequences have, only a portion of the q-nary m-sequences needed to be computed. The other portions of such sequences were then obtained by simple multiplications of elements from GF(q). The m-sequences as well as the c-m-sequences over GF(q) were random error-correcting codes. The maximum number of errors which they were able to correct depended on the minimum distance of the corresponding code, meaning that it also depended on the degree of the corresponding parity check polynomial and the Galois field being used.

In general, not all q-nary m-sequences have the one-step majority logic decodable capability but the q-nary c-m-sequences do. It is important in the sense that majority logic decoding is a well-established decoding scheme for certain classes of cyclic codes such as m-sequences. This type of decoding scheme has been investigated by many researchers and they have found some good, step-by-step

decoding procedures as well as the decoding shift registers circuits.

The c-m-sequences inherit all properties of the m-sequences and in addition they have the one-step majority logic decodable capability. Therefore it is reasonable that the q-nary c-m-sequences can be employed in communication systems where binary m-sequences are used. When information is in the form of non-binary basis, or when errors occur in bursts, it is more efficient to use non-binary codes. Since there are $q=2^s$ levels for c-m-sequences over GF(q), the bandwidth and the number of bits required to transmit such sequences are compressed by a factor of s compared with the binary case. The corresponding decoding procedures are simple because such sequences are one-step majority logic decodable and the decoding procedures have been well-established. In this thesis, special attention was given to the $GF(q^2)$ cases. Further studies may be taken upon cases with respect to other extension fields such as $GF(q^3)$, $GF(q^4)$, and so forth. Also the design of feedback shift registers circuits for generating c-m-sequences over GF(q) and/or the simplification of the one-step majority logic decoding schemes are interesting subjects for further investigation.

REFERENCES

1. Albert, A., "Fundamental Concepts of Higher Algebra," The University of Chicago Press, Chicago; Math. Rev., 20:5190, 1956.

2. Berlekamp, E.R., "Algebraic Coding Theory," McGraw-Hill, 1968.

3. Blake, I.F., and Mullin, R.C., "The Mathemetical Theory of Coding," Academic Press, 1975.

4. Clark, G.C. Jr., and Cain, J.B., "Error-Correction Coding for Digital Communications," Plenum Press, 1981.

5. Dixon, R.C., "Spread Spectrum Systems," John Wiley & Sons, 1976.

6. Gallager, R.G., "Information Theory and Reliable Communication," John Wiley & Sons, 1968.

7. Golomb, S.W., "Digital Communications With Space Applications," Prentice-Hall, 1964.

8. Golomb, S.W., "Shift Register Sequences," Holden-Day, 1967.

9. Hamming, R.W., "Error Detecting and Error Correcting Codes," Bell Systems Tech J., 29, pp. 147-160, April 1950.

10. Lin, S., "An Introduction to Error-Correcting Codes," Prentice-Hall, 1970.

11. MacWilliam, F.J., and Sloane, N.J.A., "The Theory of Error-Correcting Codes," North-Holland Publishing company, 1977.

12. Massey, J.L., "Threshold Decoding," The M.I.T. Press, Cambridge, Massachusetts, 1963.

13. McEliece, R.J., "The Theory of Information and Coding; Encyclopedia of Mathemetics and Its Applications," Addison-Wesley Publishing Company, 1977.

14. Peterson, W.W., and Weldon, E.J. Jr., "Error-Correcting Codes," 2nd edition, The M.I.T. Press, 1971.

15. Pless, V.,"Introduction to The Theory of Error-Correcting Codes," John Wiley & Sons, 1982.

16. Prange, E., "Cyclic Error-Correcting Codes in Two Symbols," AFCRC-TN-57-103, Air Force Cambridge Research Labs., Cambridge, Massachusetts, Sept. 1957.

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

18. Slepian, D., " A Class of Binary Signalling Alphabets," Bell Systems Tech. J., 35, pp. 203-234, Jan 1956.

19. Stone, H.S., "Discrete Mathemathical Structures and Their Applications," Science Research Associates, Inc., 1973.

20. Willett, M.C., "Cycle Representatives for Minimal Cycle Codes," IEEE Trans. on Info. Theory, Nov. 1975, pp 716-718.

21. Zierler, N., and Gorenstein, D., "A Class of Cyclic Linear Error-Correcting Codes in p^m Symbols," J. Soc. Ind. Appl. Math., 9, pp 107-214, June 1961.