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

CORRELATION STUDIES OF CERTAIN COMPOSITE CODES

by

Huang, Jen-Fa

A thesis submitted to the School of Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Applied Science

Department of Electrical Engineering
Faculty of Science and Engineering
The University of Ottawa
Ottawa, Canada

July, 1981

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

TABLE OF CONTENTS

ABSTRACT

Two classes of composite codes with good correlation properties are discussed in this thesis. These composite codes are combinations of suitable M-sequence codes. To examine their correlation behavior, the general aspects of code sequences are first introduced. We bring in the finite field theory, the code generation methods, the code correlation functions, and some others. With the well-established correlation bounds, pairs of preferred polynomials are searched, and Gold codes are accordingly constructed. With the introduction of dimensionality and weight distribution, the correlation spectra of generalized composite codes are derived. These two composite codes are then compared from the viewpoint of correlation and some other related concepts.

ACKNOWLEDGEMENTS

The author wishes to express his deepest gratitude to Professor S. G. S. Shiva for his guidance and encouragement throughout this research. Thanks are due to Cecillia S.L. Cheung for the many stimulating discussions and useful suggestions. Thanks are also due to Yi-Chi Shih and George H. Nehme for their helpful assistances with the computational procedures. The financial support of the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

Chapter I

INTRODUCTION

1.1 IMPORTANCE OF CODE SEQUENCES WITH GOOD CORRELATION PROPERTIES

Digital binary code sequences have been used extensively in various communication systems [5], [10], including ranging, spread spectrum, and multiple access systems. In most of these applications, sets of signals which have one or both of the following two properties are required:

i) each signal in the set is easy to distinguish from a time-shifted version of itself;

ii) each signal in the set is easy to distinguish from every other signal in the set.

The first property means that the correlation between a signal and its out-of-phase replica has to be as small as possible. This is usually called auto-correlation and is important for such applications as ranging systems, radar systems, and spread-spectrum communications systems. The second property requires that the correlation of any two different signals be always small, have better be zero value. Normally, this second function is termed cross-

correlation and is important for simultaneous ranging to several targets, multiple-terminal system identification, and code-division multiple-access communications systems.

For example, in the spread spectrum communication system [5], a binary signal u(t) consisting exclusively of positive pulse and negative pulse is modulated on a carrier, often along with other data bearing signals, and is transmitted to the receiver. In the demodulation process at the receiver, an attempt is made to synchronize a replica of u(t) with the signal modulated on the received carrier. This synchronization process is controlled by correlators whose performance depends strongly on the auto-correlation function

$$r_{u,u}(\tau) = \frac{1}{T}\int_0^T u(t)u(t+\tau)\,dt \qquad (1.1)$$

which represents the correlation of the receiver's replica with the delayed version of the transmitted signal. The quantity τ represents the error in synchronizing the received signal with the replica. In general, the value of $r_{u,u}(\tau)$ is a variable, depending on the integrating period T and the delay τ. A desirable property from the standpoint of correlator performance is that $r_{u,u}(\tau)$ be as small as possible for all T and when $\tau \neq 0$.

In the multiple access situation [23], K transmitters, using the same carrier frequency, may be simultaneously transmitting signals. The K binary signals $u_1(t)$, $u_2(t)$, ..., $u_K(t)$ used in generating the modulated K carrier waveforms are distinct. The receiver intending to listen to transmitted signal j attempts to correlate a replica of $u_j(t)$ with the received waveforms. The ability of the receiver to listen to the jth transmitter and ignore the other transmitted signals depends on the cross-correlation function

$$r_{u_j,u_k}(\tau) = \frac{1}{T}\int_0^T u_j(t)u_k(t+\tau)dt \qquad (1.2)$$

which represents a measure of the crosstalk interference between the two signals $u_j(t)$ and $u_k(t)$. Specifically when $k \neq j$, it is desirable to make $r_{u_j,u_k}(\tau)$ as small as possible for all T and τ.

The signals employed in the kinds of applications mentioned above are usually required to be periodic. This is primarily because of the simplifications in system implementation that typically result from the use of periodic signals. Consequently, we restrict our attention to periodic signals throughout this thesis. In order to simplify the presentation, we consider only those sets of signals which have the property that for some T, u(t) =

$u(t+T)$ for all t and for each signal u in the set. The most common example is a set of signals of common period T; however, all that is required is that T be an integer multiple of the period of each signal in the set.

In addition, due in part to the relative simplicity of their generation, the signals of interest for most applications are periodic signals which consist of sequences of elemental time-limited pulses. These pulses are all of the same shape, so that the signal can be written as

$$u(t) = \sum_{i=-\infty}^{\infty} u_i \varphi(t-iT_c), \qquad (1.3)$$

where $\varphi(t)$ is the basic pulse waveform and T_c is the time duration of this pulse. If $u(t) = u(t+T)$ for all t, then T must be a multiple of T_c, and the sequence u must be periodic with a period which is a divisor of $n = T/T_c$.

Properties i) and ii) require distinguishability of two periodic signals $u(t)$ and $v(t+\tau)$ for all $\tau \in [0,T]$ if $u(t)$ and $v(t)$ are different signals, and for all $\tau \in (0,T)$ if $u(t)$ and $v(t)$ are the same. Consequently, it is the magnitude of the correlation function

$$r_{u,v}(\tau) = \frac{1}{T}\int_0^T u(t)v(t+\tau)dt \qquad (1.4)$$

that is of interest. With u(t) given by (1-3) and with v(t) given by

$$v(t) = \sum_{i=-\infty}^{\infty} v_i \varphi(t-iT_c). \tag{1.5}$$

Then if $\tau = lT_c$, the parameter $r_{u,v}(\tau)$ of (1-4) can generalize to

$$r_{u,v}(\tau) = \frac{\lambda}{n} \sum_{i=0}^{n-1} u_i v_{i+l}, \tag{1.6}$$

where the constant λ is

$$\lambda = \frac{1}{T_c} \int_0^{T_c} \varphi^2(t)dt. \tag{1.7}$$

For example, if $\varphi(t)$ is the unit amplitude rectangular pulse of duration T_c which starts at $t = 0$, then we have $\lambda = 1.0$. Equ.(1-6) is the constant λ of (1-7) multiplied by the normalized inner product of code vectors $(u_0, u_1, \ldots, u_{n-1})$ and $(v_l, v_{l+1}, \ldots, v_{l+n-1})$. Since code sequence v has a period which divides n, then

$$(v_l, v_{l+1}, \ldots, v_{l+n-1})$$
$$= (v_l, v_{l+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{l-1}). \tag{1.8}$$

The right-hand side of (1-8) is the l-th cyclic shift of the original code vector $(v_0, v_1, \ldots, v_{n-1})$.

The above discussion motivates the consideration of the periodic correlation function for code sequences u and v which is defined by

$$\theta_{u,v}(l) = \frac{1}{n} \sum_{i=0}^{n-1} u_i v_{i+l}. \qquad (1.9)$$

Compare (1-6) and (1.9), we see that $r_{u,v}(\tau) = \lambda \cdot \theta_{u,v}(l)$ whenever $\tau = lT_c$. Hence the correlation function of the continuous-time signals u(t) and v(t) of (1-3) and (1-5) is determined by that of their corresponding code sequences. Due to this fact, the signal design problem in the previously mentioned communication systems is equivalent to the problem of finding sets of code sequences with the following two properties:

1') for each code sequence u in the set, the auto-correlation $|\theta_{u,u}(l)|$ is small for $1 \leq l \leq n-1$;

2') for each pair of code sequences u and v, the cross-correlation $|\theta_{u,v}(l)|$ is small for all l.

Code sequences that satisfy these demands will play a deterministic role for the successful system performance. Unfortunately, most of the known codes fail to satisfy both of the above listed requirements.

For instance, the maximal-length code sequences of period $n = 2^m-1$ generated from m-stage linear feedback shift registers have been known for a long time and are widely used in many areas of communication systems. One of their key features is their 2-level periodic auto-correlation function,

$$\theta(l) = \begin{cases} 1 , & \text{for } l = 0; \\ -1/n, & \text{for } 1 \le l \le n-1, \end{cases} \qquad (1.10)$$

which is the best possible auto-correlation function for a binary code of length $n = 2^m-1$. It is this ideal periodic auto-correlation property that was exploited in most of the early applications of maximal-length codes. However, if we make a close study on the cross-correlation properties of maximal-length codes, we find the situation somewhat disappointing. Except for some carefully chosen sets, the peak cross-correlation between any two maximal-length codes are not only randomly distributed but also relatively high valued. Higher cross-correlation value means poorer receiver's ability to recognize the proper point of code synchronization. Therefore, in view of their correlation properties, maximal-length codes cannot be said to be "best" in the above mentioned spread spectrum or multiple access applications.

Though maximal-length codes cannot be bettered in some of systems applications, we must admit that composite codes generated by combination of linear maximal-length codes should be considered, and accordingly, their correlation properties and some other important characteristics have to be investigated. It is this reason which motivated the author to undertake this study, and fortunately, as we shall see, the composite codes are much better from their correlation point of view than the maximal-length codes and therefore are well suited for the above mentioned spread-spectrum and multiple-access applications.

1.2 HISTORICAL NOTES OF COMPOSITE CODES AND OUTLINE OF THE THESIS

The studies of composite codes can be traced back to the 1960s. At that time, binary maximal length shift register sequences had been well investigated and some results on their structural properties, correlation functions, methods of generation, and applications to various electronic systems problems had been obtained. But with some new applications such as those previously mentioned, a few further requirements were also needed, like larger code set, rapid code acquisition, small cross-correlation and out-of-phase auto-correlation values, and so on. Faced with these requirements, maximal-length code sequences soon lost their optimal status. Hence, some researchers turned their

attentions to the construction of suitable code sequences which satisfy some specified auto-correlation and cross-correlation criterions. Among these, the composite codes made up of the well-known maximal-length codes are our main concern. This is because of their easy construction and implementation in the practical applications, and also due to their easy analysis on the algebraic structure in connection with maximal-length codes.

The first composite code was proposed by Titsworth [28] in 1964 to be used in ranging systems. By the use of a majority-logic combination function, several maximal-length "component" codes with relatively prime length are encoded into a transmitted signal. The receiver correlates the delayed return signal with different Boolean combinations of delayed replicas of the components to determine separately the time delay of each component code. From these delays, the total delay and hence the range are computed in a relatively short time.

Gold [8] devised another class of composite codes in 1967 for spread spectrum multiplexing applications, which have been named Gold code by many following researchers in honor of Gold's contribution. The Gold code is a subset of the famous BCH codes and it contains a large number of useful codes which are generated by a particular generator

configuration and possess well-bounded auto-correlation and cross-correlation values.

By utilizing Karnaugh chart and probabilities, Easterling [10, Chapter 5] investigated a family of combination waveforms on both of its correlation functions and power spectra. From the computational results, Easterling pointed out that the correlation functions of his developed composite codes have minor peaks for all cyclic shift positions which are divisible by the period of either component codes. Unfortunately, there are no explicit indications on how large the peak values are and how often the peaks occur.

Milstein and Ragonetti [19], [20] re-examined Titsworth's composite codes and derived out some statistical properties on both partial and full correlations. They also suggested the use of that class of composite codes in the environment where a relatively small number of users are in need of long but rapidly acquirable codes.

In general, correlation values are related with minimum distances or weights of the code. It is thus not surprising that the weight distribution of a code can be converted into the equivalent correlation spectra. For example, in the presentation of [25], Sarwate and Pursley translated

Kasami's weight distribution formulas [13] of BCH codes into results on correlation spectra of Gold codes. In the same fashion, McEliece [16] derived the correlation properties of irreducible cyclic codes from codes' minimum weights developed earlier by himself and Baumert [2].

For judging whether a large code set is optimal in view of correlation properties, Welch [31] established lower bounds on how small the cross-correlation and the auto-correlation can simultaneously be. Sarwate [24] modified and expanded Welch's proof and concluded that the ideal auto-correlation and cross-correlation functions cannot exist simultaneously but have to suffer trade-offs with each other. According to their results, if the code set X contains K words of period n, then the maximum periodic cross-correlation magnitude θ_c and the maximum periodic out-of-phase auto-correlation magnitude θ_a, defined respectively by

$$\theta_c = \max \{ |\theta_{u,v}(l)| : U,V \in X,\ U \neq V,\ 0 \leq l \leq n-1 \} \tag{1.11}$$

and

$$\theta_a = \max \{ |\theta_{u,u}(l)| : U \in X,\ 1 \leq l \leq n-1 \} \tag{1.12}$$

will relate through the inequality

$$n\theta_c^2 + \left(\frac{n-1}{K-1}\right)\theta_a^2 \geq 1 \tag{1.13}$$

which provides a lower bound on one of the maxima if the value of the other is specified.

In the context of the preceding discussions, we discuss in this thesis several aspects of composite codes, namely, generation methods, correlation spectra, correlation bounds, and some other related properties. Generally, there are two types of correlation functions: the periodic correlation functions which take full period of codewords for correlation evaluation, and aperiodic correlation functions which consider any nonperiodic code bits for correlation calculation. Also, there are several types of code sequences which can be constructed, like complex code sequences [22], [25], polyphase code sequences [4], [29], and irreducible cyclic codes [15], [16], etc. In this thesis, only periodic correlation properties of composite codes made up of M-sequence codes will be studied.

In Chapter II, we discuss the general aspects of linear cyclic code sequences. First, in Section 2.1, the basic elements of finite field theory are given. This introduces the important concepts of Galois fields, cyclotomic cosets, and minimum polynomials. Based on these irreducible minimum polynomials, linear feedback shift registers can be suitably connected to generate the desired code sequences. In Section 2.2 we present such code sequences generating

configurations. The general properties of binary (n,k) cyclic codes are then examined. The terms minimum distances, weight distributions, periodic correlation functions and some others required for the development of the following sections are also defined. In Section 2.3, as a basis to the correlation study of our composite codes, the essential characteristics of M-sequence codes are outlined.

In Chapter III, we come to the main points of the thesis. By combining pair of preferred M-sequence codes, sets of Gold codes are generated. With respect to this class of codes we examine the number of codes and their correlation properties. This is then followed by the introduction of generalized composite codes, which are combined from pairwise relatively prime length M-sequence codes. Compared with Gold codes, the number of codes and correlation functions of generalized composite codes are fully discussed. The relative cyclic shift positions for any specified correlation values are also derived.

Finally in Chapter IV, conclusions and recommendations for further work are presented based on the results of this study.

Chapter II

GENERAL ASPECTS OF CYCLIC CODE SEQUENCES

In this chapter, the basic elements of finite field theory required for describing code sequences are first quoted. Also we introduce the important concepts of cyclotomic cosets and minimal polynomials. Based on these irreducible polynomials, linear feedback shift registers can be suitably connected to generate the desired code sequences. From shift register code generator output, the general properties of (n,k) cyclic codes are examined. The terms minimum distance, weight distribution, periodic correlation functions (auto-correlation and cross-correlation) and some others are also defined. Finally, as a basis to the correlation study of our composite codes, the essential characteristics of M-sequence codes are outlined.

2.1 BASIC FINITE FIELD THEORY

Just like those of integer numbers, polynomials can also be added and multiplied in the usual way. Now consider a polynomial P(X) of degree m with coefficients from binary field GF(2) = {0,1} :

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

If the coefficient p_m of the highest power of X is 1, then polynomial P(X) is called monic. If the polynomial P(X) contains no binary polynomial factors except itself and 1, then P(X) is called irreducible over GF(2). A polynomial P(X) is said to have exponent ν if it divides $1 + X^{\nu}$ and not any $1 + X^{\mu}$ for any $\mu < \nu$.

Choose a suitable irreducible polynomial P(X) of degree m such that for a symbol α other than any field element of GF(2), $P(\alpha) = 0$, then the set of all polynomials in α of degree $\leq m-1$ and coefficients from GF(2), with calculations performed modulo $P(\alpha)$, forms a field of order 2^m. This field is an extension field of GF(2), and is denoted by $GF(2^m)$.

The element α is called a primitive element of the field $GF(2^m)$. In general, any element of $GF(2^m)$ whose powers generate all the non-zero elements of $GF(2^m)$ is said to be primitive. The irreducible polynomial P(X) of degree m which has a primitive element of $GF(2^m)$ as a root is called a primitive polynomial.

An example of the field $GF(2^4)$ generated by $P(\alpha) = 1+\alpha+\alpha^4 = 0$ is shown in Figure 2.1.1. Note that the field elements can be written in several different ways.

As a power of α	As a polynomial	As a 4-tuple
0	0	0 0 0 0
1	1	0 0 0 1
α	α	0 0 1 0
α^2	α^2	0 1 0 0
α^3	α^3	1 0 0 0
α^4	$1 + \alpha$	0 0 1 1
α^5	$\alpha + \alpha^2$	0 1 1 0
α^6	$\alpha^2 + \alpha^3$	1 1 0 0
α^7	$1 + \alpha + \alpha^3$	1 0 1 1
α^8	$1 + \alpha^2$	0 1 0 1
α^9	$\alpha + \alpha^3$	1 0 1 0
α^{10}	$1 + \alpha + \alpha^2$	0 1 1 1
α^{11}	$\alpha + \alpha^2 + \alpha^3$	1 1 1 0
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	1 1 1 1
α^{13}	$1 + \alpha^2 + \alpha^3$	1 1 0 1
α^{14}	$1 + \alpha^3$	1 0 0 1

Figure 2.1.1: $GF(2^4)$ generated by logic $1+\alpha+\alpha^4=0$.

The minimal polynomial over GF(2) of element $\alpha \in GF(2^m)$ is the lowest degree monic polynomial $M(X)$ such that $M(\alpha)=0$. This minimal polynomial possesses the following important properties:

M1) $M(X)$ is irreducible;

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

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

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

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

M6) α and α^2 have the same minimal polynomial.

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

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

$$\{ s, 2s, 2^2 s, \ldots, 2^{m_s-1} s \},$$

where m_s is the smallest positive integer such that $2^{m_s} s = s \bmod (2^m-1)$.

Our notation is that if s is the smallest number in the coset, the coset is called C_s. The subscript s is called coset leader modulo (2^m-1), or simply coset leader. Accordingly, we shall let $M_s(X)$ be the minimal polynomial of $\alpha^s \in GF(2^m)$. Of course, by property M6),

$$M_{2s}(X) = M_s(X). \qquad (2.1.1)$$

Table 2.1.1 lists the cyclotomic cosets modulo 2^4-1, 2^5-1, 2^6-1 and 2^7-1, respectively.

Table 2.1.1

Cyclotomic cosets modulo 15, 31, 63, and 127.

mod 15

$C_0 = \{0\}$
$C_1 = \{1, 2, 4, 8\}$
$C_3 = \{3, 6, 12, 9\}$
$C_5 = \{5, 10\}$
$C_7 = \{7, 14, 13, 11\}$

mod 31

$C_0 = \{0\}$
$C_1 = \{1, 2, 4, 8, 16\}$
$C_3 = \{3, 6, 12, 24, 17\}$
$C_5 = \{5, 10, 20, 9, 18\}$
$C_7 = \{7, 14, 28, 25, 19\}$
$C_{11} = \{11, 22, 13, 26, 21\}$
$C_{15} = \{15, 30, 29, 27, 23\}$

mod 63

$C_0 = \{0\}$
$C_1 = \{1, 2, 4, 8, 16, 32\}$
$C_3 = \{3, 6, 12, 24, 48, 33\}$
$C_5 = \{5, 10, 20, 40, 17, 34\}$
$C_7 = \{7, 14, 28, 56, 49, 35\}$
$C_9 = \{9, 18, 36\}$
$C_{11} = \{11, 22, 44, 25, 50, 37\}$
$C_{13} = \{13, 26, 52, 41, 19, 38\}$
$C_{15} = \{15, 30, 60, 57, 51, 39\}$
$C_{21} = \{21, 42\}$
$C_{23} = \{23, 46, 29, 58, 53, 43\}$
$C_{27} = \{27, 54, 45\}$
$C_{31} = \{31, 62, 61, 59, 55, 47\}$

mod 127

$C_0 = \{0\}$
$C_1 = \{1, 2, 4, 8, 16, 32, 64\}$
$C_3 = \{3, 6, 12, 24, 48, 96, 65\}$
$C_5 = \{5, 10, 20, 40, 80, 33, 66\}$
$C_7 = \{7, 14, 28, 56, 112, 97, 67\}$
$C_9 = \{9, 18, 36, 72, 17, 34, 68\}$
$C_{11} = \{11, 22, 44, 88, 49, 98, 69\}$
$C_{13} = \{13, 26, 52, 104, 81, 35, 70\}$
$C_{15} = \{15, 30, 60, 120, 113, 99, 71\}$
$C_{19} = \{19, 38, 76, 25, 50, 100, 73\}$
$C_{21} = \{21, 42, 84, 41, 82, 37, 74\}$
$C_{23} = \{23, 46, 92, 57, 114, 101, 75\}$
$C_{27} = \{27, 54, 108, 89, 51, 102, 77\}$
$C_{29} = \{29, 58, 116, 105, 83, 39, 78\}$
$C_{31} = \{31, 62, 124, 121, 115, 103, 79\}$
$C_{43} = \{43, 86, 45, 90, 53, 106, 85\}$
$C_{47} = \{47, 94, 61, 122, 117, 107, 87\}$
$C_{55} = \{55, 110, 93, 59, 118, 109, 91\}$
$C_{63} = \{63, 126, 125, 123, 119, 111, 95\}$

Corresponding to each cyclotomic coset C_s, the minimal polynomial $M_s(X)$ of $\alpha^s \in GF(2^m)$ can be found by various methods such as Berlekamp's [3]. However, these have been compiled for most cases of practical interest and there are available tables of binary irreducible

polynomials like those edited by Peterson and Weldon [21, Appendix C]. For example, with m = 6, the table in [21] contains the entry

DEGREE 6 1 103F 3 127B 5 147H 7 111A 9 015
11 155E 21 007

Here, following each coset leader s, the minimal polynomial is given in an octal representation. For elements belonging to the same cyclotomic coset, the corresponding minimal polynomials are the same, so only those of coset leaders are listed in the table. Also, of any pair consisting of a polynomial and its reciprocal, only one is listed. In the above given polynomial entries, the first entry is supposed to have the primitive element α as a root and is the one used in defining the discussed field $GF(2^6)$. Each digit of the octal notation can be transformed into three binary digits equivalent which are the coefficients of the polynomial, with the high-order coefficients at the left. The letters E, F, and H mean (among other things) that the polynomials (103), (147), and (155) are primitive while A and B indicate nonprimitive polynomials. Therefore, the minimal polynomial of $\alpha \in GF(2^6)$ is the primitive polynomial (103) used in defining the discussed field $GF(2^6)$. The binary equivalent of (103) is 001000011, and hence the polynomial representation is $1+X+X^6$. Similarly, the minimal

polynomial of $\alpha^3 \in GF(2^6)$ is the nonprimitive (127) which has binary equivalent 001010111, so the polynomial representation is $1+X+X^2+X^4+X^6$.

There is no entry corresponding to α^{13}, so the minimal polynomial of α^{13} must be some reciprocal polynomial of the above listed polynomial entries. In general, the reciprocal polynomial of $P(X) = p_0 + p_1X + \ldots + p_{m-1}X^{m-1} + p_mX^m$ is

$$P^*(X) = X^{\deg P(X)}P(X^{-1}) = X^mP(X^{-1})$$

$$= p_m + p_{m-1}X + \ldots + p_1X^{m-1} + p_0X^m \quad (2.1.2)$$

obtained by reversing the order of the coefficients. The roots of the reciprocal polynomial are the reciprocals of roots of the original polynomial. The reciprocal of an irreducible polynomial is also irreducible, and the reciprocal of a primitive polynomial is primitive. Consequently, if α^s has minimal polynomial $M_s(X)$, we know immediately that α^{-s} has minimal polynomial $M_s^*(X)$ = reciprocal polynomial of $M_s(X)$.

Now, to find the minimal polynomial of $\alpha^{13} \in GF(2^6)$, let's check the roots of the reciprocal polynomial $M_{13}^*(X)$ of $M_{13}(X)$. These roots are

$$\alpha^{-13} = \alpha^{50},\ \alpha^{-26} = \alpha^{37},\ \alpha^{-52} = \alpha^{11},$$

$$\alpha^{-104} = \alpha^{-41} = \alpha^{22},\ \alpha^{-82} = \alpha^{-19} = \alpha^{44},\ \alpha^{-38} = \alpha^{25}.$$

The minimal polynomial of α^{11} is listed in the given entry as (155), or $M_{11}(X) = 1+X^2+X^3+X^5+X^6$. The minimal polynomial of α^{13} is therefore the reciprocal polynomial of this, or $M_{13}(X) = M^*_{11}(X) = 1+X+X^3+X^4+X^6$.

Following the above discussion, the minimal polynomials of elements $\alpha^j \in GF(2^6)$, $0 \le j < 2^6-1$, can all be derived. Figure 2.1.2 gives the minimal polynomial corresponding to each element in $GF(2^6)$ defined by $1+\alpha+\alpha^6=0$.

Elements	Minimal Polynomials
0	$M_{-\infty}(X)=X$
1	$M_0(X)=1+X$
$\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}\}$	$M_1(X)=1+X+X^6$
$\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{33}\}$	$M_3(X)=1+X+X^2+X^4+X^6$
$\{\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34}\}$	$M_5(X)=1+X+X^2+X^5+X^6$
$\{\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35}\}$	$M_7(X)=1+X^3+X^6=M^*_7(X)$
$\{\alpha^9, \alpha^{18}, \alpha^{36}\}$	$M_9(X)=1+X^2+X^3$
$\{\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37}\}$	$M_{11}(X)=1+X^2+X^3+X^5+X^6$
$\{\alpha^{13}, \alpha^{26}, \alpha^{52}, \alpha^{41}, \alpha^{19}, \alpha^{38}\}$	$M_{13}(X)=1+X+X^3+X^4+X^6=M^*_{11}(X)$
$\{\alpha^{15}, \alpha^{30}, \alpha^{60}, \alpha^{57}, \alpha^{51}, \alpha^{39}\}$	$M_{15}(X)=1+X^2+X^4+X^5+X^6=M^*_3(X)$
$\{\alpha^{21}, \alpha^{42}\}$	$M_{21}(X)=1+X+X^2=M^*_{21}(X)$
$\{\alpha^{23}, \alpha^{46}, \alpha^{29}, \alpha^{58}, \alpha^{53}, \alpha^{43}\}$	$M_{23}(X)=1+X+X^4+X^5+X^6=M^*_5(X)$
$\{\alpha^{27}, \alpha^{54}, \alpha^{45}\}$	$M_{27}(X)=1+X+X^3=M^*_9(X)$
$\{\alpha^{31}, \alpha^{62}, \alpha^{61}, \alpha^{59}, \alpha^{55}, \alpha^{47}\}$	$M_{31}(X)=1+X^5+X^6=M^*_1(X)$

Figure 2.1.2: Minimal polynomials of elements in $GF(2^6)$ defined by $1+\alpha+\alpha^6=0$.

The exponent to which the minimal polynomial $M_s(X)$ belongs can be found as follows:

$$e = \frac{2^m-1}{GCD(2^m-1,s)}, \qquad (2.1.3)$$

where GCD refers to the greatest common divisor henceforth simply written as (p,q). Obviously, $M_s(X)$ has maximum exponent $e = 2^m-1$ if and only if $(2^m-1,s) = 1$, and in this case $M_s(X)$ is a primitive polynomial of degree m. The number of binary irreducible polynomials of degree m is generally given by [11]

$$\psi_2(m) = \frac{1}{m} \sum_{b|m} 2^b \mu(\frac{m}{b}), \qquad (2.1.4)$$

where the sum is over all divisors b of m, and $\mu(a)$ is the Mobius function:

$$\mu(a) = \begin{cases} 1 & , \text{ if } a=1; \\ 0 & , \text{ if a has any square factor;} \\ (-1)^r & , \text{ if } a=p_1p_2\cdots p_r, \text{ where } p_1, p_2, \cdots, p_r \text{ are distinct primes.} \end{cases} \qquad (2.1.5)$$

From (2.1.3), it can be observed that not all irreducible polynomials of degree m have maximum exponent. The number of binary polynomials of degree m which have maximum exponent is given by

$$\lambda_2(m) = \frac{\phi(2^m-1)}{m}, \qquad (2.1.6)$$

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

Table 2.1.2 gives a tabulation of $\psi_2(m)$ and $\lambda_2(m)$.

Table 2.1.2

The tabulation of $\psi_2(m)$ and $\lambda_2(m)$.

m	2^m-1	$\psi_2(m)$	$\lambda_2(m)$
1	1	2	1
2	3	1	1
3	7	2	2
4	15	3	2
5	31	6	6
6	63	9	6
7	127	18	18
8	255	30	16
9	511	56	48
10	1,023	99	60
11	2,047	186	176
12	4,095	335	144
13	8,191	630	630
14	16,383	1,161	756
15	32,767	2,182	1,800
16	65,535	4,080	2,048

2.2 CODE SEQUENCES GENERATION AND THEIR GENERAL PROPERTIES

Linear code sequences can be easily generated from linear feedback shift register circuits, so they are sometimes called linear shift register sequences. A k-stage shift register is a device consisting of k consecutive memory elements. The contents of each memory element shift to the

next memory element down the line in time to the regular beat of a clock or to other timing device. A feedback term is computed as a logical function of the contents of k memory elements and fed back into the leftmost memory element of the shift register. A block diagram of the shift register with the feedback mechanism is shown in Figure 2.2.1, where the M's are memory elements. Note that the memory cells are numbered from right to left.

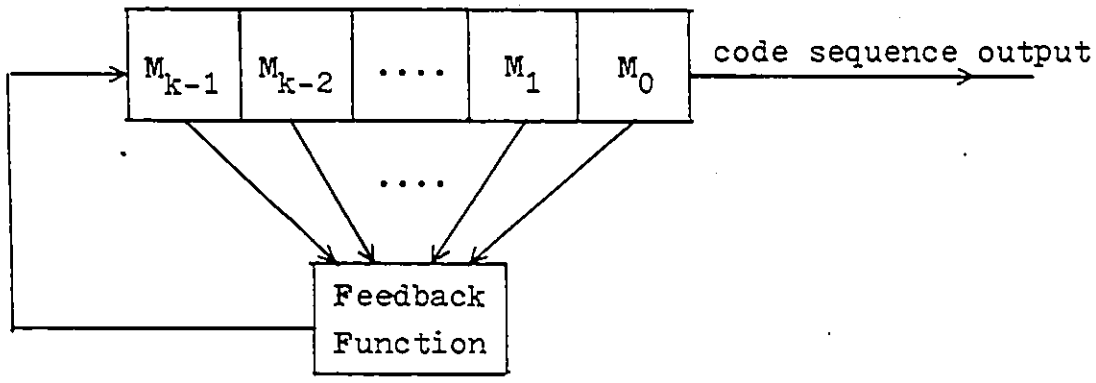


Figure 2.2.1: Shift register feedback mechanism.

Now, suppose an initial binary k-tuple $(i_0, i_1, \ldots, i_{k-1})$ is loaded on the memory cells of the k-stage shift register. Let $h(X)=h_0+h_1X+\ldots+h_{k-1}X^{k-1}+h_kX^k$ denote a binary polynomial of degree k where $h_0=h_k=1$ and the other h_i's take on values 0 and 1. Corresponding to h(X), if we take linear

feedback function of the shift register circuit to be

$$f(h_i,M_i) = h_0M_0 \oplus h_1M_1 \oplus \cdots \oplus h_{k-1}M_{k-1}, \qquad (2.2.1)$$

where $M_i \in \{0,1\}$, i=0, 1, 2, ..., k-1, is the content of the (i+1)th memory cell of the shift register, and $\oplus$ is the modulo-2 addition or the EXCLUSIVE-OR logic operation, then the successive bits of code sequence v generated from the above shift register circuit will satisfy the following recurring relation for all integers j,

$$h_0v_j \oplus h_1v_{j+1} \oplus \cdots \oplus h_{k-1}v_{j+k-1} \oplus h_kv_{j+k} = 0. \qquad (2.2.2)$$

From the above equation, and using the fact that $h_k=1$, we obtain

$$v_{j+k} = h_0v_j \oplus h_1v_{j+1} \oplus \cdots \oplus h_{k-1}v_{j+k-1}. \qquad (2.2.3)$$

The polynomial h(X) associated with (2.2.1) is the recursion or characteristic polynomial of code sequence v. The k-stage linear feedback shift register used for generating v has a feedback tap connected to the (i+1)th cell if $h_i=1$, $0 \le i \le k-1$. Since $h_0=1$, there is always such a connection for the first cell. For example, the shift register in Figure 2.2.2(a) corresponds to $h(X)=1+X^4+X^5$ while that in Figure 2.2.2(b) corresponds to $h(X)=1+X+X^2+X^4+X^5$.

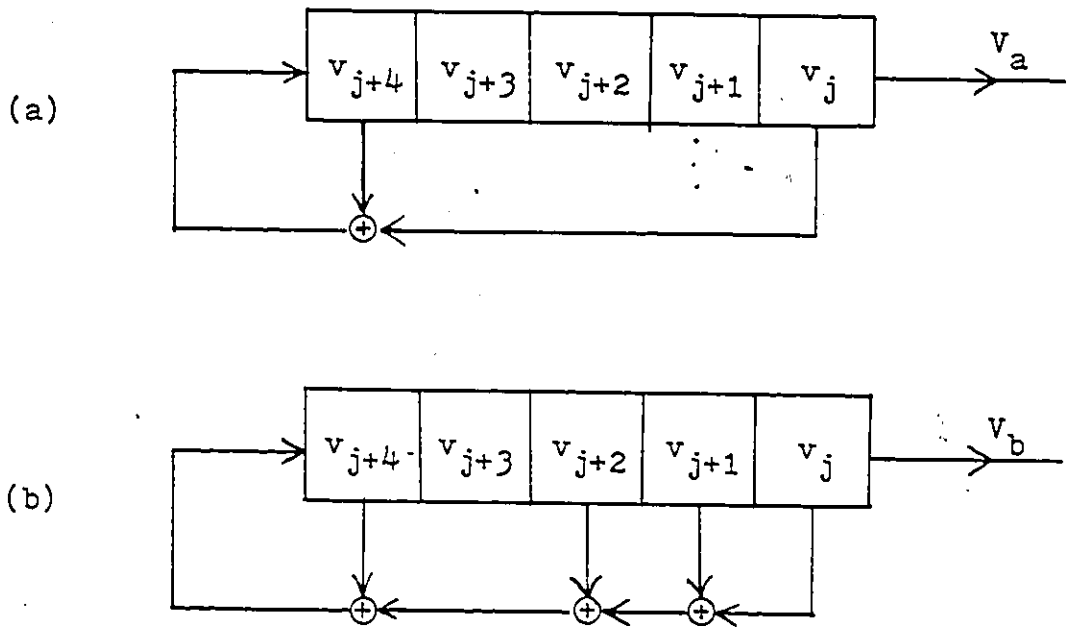


Figure 2.2.2: Linear feedback shift register code generators. (a) corresponds to $h(X)=1+X^4+X^5$. (b) corresponds to $h(X)=1+X+X^2+X^4+X^5$.

To make clear how code sequences can be generated from linear feedback shift registers, let's consider the above generator configurations of Figures 2.2.2(a) and 2.2.2(b), both with the initial memory setting of the 5-tuple (1,1,1,1,1). Once started by the regular beat of the timing clock, the successive states of the shift registers of Figures 2.2.2(a) and 2.2.2(b) will respectively be like those shown in Figures 2.2.3(a) and 2.2.3(b). The output sequences are the respective right-hand column of the lists of states. Note that in either shift register circuit if a different nonzero initial state is used, it is still one of the states the shift register goes through, and the new output sequence is just a cyclic shift of the original

sequence we presented. Hence the same set of binary linear sequences is obtained from any nonzero starting state.

(a)

1	1	1	1	1
0	1	1	1	1
1	0	1	1	1
0	1	0	1	1
1	0	1	0	1
0	1	0	1	0
0	0	1	0	1
1	0	0	1	0
1	1	0	0	1
0	1	1	0	0
0	0	1	1	0
0	0	0	1	1
1	0	0	0	1
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0
1	1	1	1	1
0	1	1	1	1

(repeats)

(b)

1	1	1	1	1
0	1	1	1	1
1	0	1	1	1
0	1	0	1	1
0	0	1	0	1
0	0	0	1	0
1	0	0	0	1
0	1	0	0	0
0	0	1	0	0
1	0	0	1	0
0	1	0	0	1
1	0	1	0	0
0	1	0	1	0
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1
0	0	1	1	0
0	0	0	1	1
0	0	0	0	1
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
1	0	0	1	1
1	1	0	0	1
0	1	1	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1
0	1	1	1	1

(repeats)

Figure 2.2.3: Successive states and output sequences from shift registers of Figure 2.2.2.

From Figures 2.2.3(a) and 2.2.3(b) we can find that the output sequence generated by the recursion polynomial $h(X)=1+X^4+X^5$ has a period of 21 bits while that generated by $h(X)=1+X+X^2+X^4+X^5$ has 31 bits. These are the typical examples of nonmaximal-length code sequence and maximal-length code sequence. Obviously, the difference in these two sequences is due to their generation by two different recursion polynomials. Taking a closer look we can find out the polynomial $h(X)=1+X^4+X^5$ is not irreducible, $1+X^4+X^5$ can be factored into $(1+X+X^2)(1+X+X^3)$ and has exponent $3 \times 7 = 21$, while the polynomial $h(X)=1+X+X^2+X^4+X^5$ is a really irreducible polynomial with exponent $2^5-1=31$.

In the above we have presented the practical generation scheme of binary linear (n,k) cyclic codes, where, 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 according to a polynomial of degree k and exponent n, an n-tuple $(v_0, v_1, \ldots, v_{n-1})$ is generated. This n-tuple consists of the unaltered k information bits followed by (n-k) linear combinations of the information bits. It is referred to as a code word or code vector of the (n,k) cyclic code V.

An (n,k) cyclic code is a linear block code with the property that any cyclic permutation of the digits in a code word is also a cyclic code word. That is, if

$$v = (v_0, v_1, \ldots, v_{n-1}) \qquad (2.2.4)$$

is a cyclic code word, then any cyclic shift of the digits, such as

$$(v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{n-i-2}, v_{n-i-1}), \qquad (2.2.5)$$

is also a cyclic code word for all i.

The sum of two cyclic code words $u=(u_0, u_1, \ldots, u_{n-1})$ and $v=(v_0, v_1, \ldots, v_{n-1})$ is also a cyclic code word and is given by

$$\begin{aligned} u + v &= (u_0, u_1, \ldots, u_{n-1}) + (v_0, v_1, \ldots, v_{n-1}) \\ &= (u_0+v_0, u_1+v_1, \ldots, u_{n-1}+v_{n-1}). \end{aligned} \qquad (2.2.6)$$

Multiplication of code vector v by field element a is defined as

$$\begin{aligned} a \cdot v &= a \cdot (v_0, v_1, \ldots, v_{n-1}) \\ &= (av_0, av_1, \ldots, av_{n-1}). \end{aligned} \qquad (2.2.7)$$

For the binary case the above addition and multiplication operations are performed in modulo-2 arithmetic. That is, they follow the following truth tables:

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

$\odot$	0	1
0	0	0
1	0	1

Corresponding to code vector v, we have a code polynomial representation given by

$$V(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}. \qquad (2.2.8)$$

In general,

$$X^iV(X) \bmod (1+X^n)$$

$$= v_{n-i} + v_{n-i+1}X + \ldots + v_{n-1}X^{i-1} + v_0X^i + v_1X^{i+1} +$$

$$\ldots + v_{n-i-2}X^{n-2} + v_{n-i-1}X^{n-1} \qquad (2.2.9)$$

is the code polynomial corresponding to codeword of (2.2.5), Note that $X^iV(X)$, i=0, 1, 2, ..., n-1, are the phases of the cyclic code V; they are cyclically equivalent, that is, they are cyclic shifts of each other. Hereafter we shall use the terms code word, code vector, and code polynomial interchangeably.

The previously mentioned information k-tuple $(i_0, i_1, \ldots, i_{k-1})$ can be expressed in polynomial form by

$$I(X) = i_0 + i_1X + \ldots + i_{k-1}X^{k-1}. \qquad (2.2.10)$$

The encoding of this information polynomial I(X) into the corresponding code polynomial V(X) is through the following relationship:

$$V(X) = g(X)I(X) \bmod (1+X^n), \qquad (2.2.11)$$

where g(X) is referred to as the generator polynomial of the cyclic code V. By connecting this generator polynomial with the above mentioned recursion polynomial we have

$$g(X) = \frac{1+X^n}{h(X)} \quad . \tag{2.2.12}$$

Obviously, the generator polynomial g(X) is a factor of $1+X^n$ and has degree (n-k) and exponent n.

Since a code word can be expressed in terms of either generator polynomial or recursion polynomial, if no confusion arises, we shall alternatively say that the cyclic code V is generated by the generator polynomial g(X) or by the recursion polynomial h(X). In either case, bear in mind that: when we say V(X) is generated by g(X), Equ.(2.2.11) is applied, but if we say V(X) is generated by h(X), then h(X) is the polynomial based to connect the shift register feedback circuit.

In general, if $V(X)=v_0+v_1X+\ldots+v_{n-2}X^{n-2}+v_{n-1}X^{n-1}$ is a code polynomial of length n generated by the generator polynomial g(X) (the recursion polynomial h(X)), then the reciprocal polynomial of V(X), $V^*(X) = v_{n-1}+ v_{n-2}X+ \ldots+ v_1X^{n-2}+ v_0X^{n-1}$, is generated by $g^*(X)$ ($h^*(X)$), the reciprocal polynomial of g(X) (h(X)).

An important concept in the study of linear codes is the (Hamming) distance between two code words. This is defined as the number of positions in which corresponding elements are different. A related concept is the (Hamming) weight of a code word which is defined as the number of non-zero elements in the code word. For example, the distance between the 7-tuples $v_i=(1,1,0,1,0,0,1)$ and $v_j=(1,0,1,0,0,1,1)$ is $dis(v_i,v_j)=4$ and both have weight $|v_i|=|v_j|=4$. For any cyclic code V, it is easily shown that $dis(v_i,v_j) = |v_i+v_j|$ for $v_i,v_j \in V$. Also, it can be shown that the minimum distance, $d = \min_{i,j}\{dis(v_i,v_j)\}$, is equal to the minimum weight of the nonzero vectors of the code.

Associated with Hamming distances and weights is the weight distribution of code words. This is defined as the number of code vectors of each possible weight in the code. Generally, if $w_1, w_2, \ldots, w_t$ are the possible weights of code words in an (n,k) binary cyclic code, and $N(w_i)$, $i = 1, 2, \ldots, t$, is the corresponding number of code words having weight w_i, then

$$\sum_{i=1}^{t} N(w_i) = 2^k. \qquad (2.2.13)$$

The periodic correlation functions (auto-correlation and cross-correlation) of binary cyclic codes are established on the assumption that the correlating codewords contain the

same number of binary symbols. Conventionally, the normalized correlation function $\theta_{u,v}(l)$ of two n-dimensional codewords U(X) and V(X) is given by

$$\theta_{u,v}(l) = \frac{1}{n} \sum_{i=1}^{n} u_i v_{i+l}; \qquad (2.2.14)$$

but in most practical communication applications, it is often desirable to use binary vectors which do not consist specifically of 0 and 1. However, it is usually desirable to preserve the correlation properties present in the {0,1} representation. This can be done by defining correlation more generally. Thus if we let

$$\theta_{u,v}(l) = \frac{A_l - D_l}{A_l + D_l}, \qquad (2.2.15)$$

where A_l is the number of places in which the words U(X) and $X^l V(X)$ agree and D_l is the number of places in which the words U(X) and $X^l V(X)$ differ, then we get a 'generalized correlation' definition which coincides with the one previously mentioned when the binary symbols are 0 and 1. Since $A_l = n - D_l$, the definition of (2.2.15) is equivalent to

$$\theta_{u,v}(l) = \frac{n - 2 \cdot dis(U(X), X^l V(X))}{n}$$

$$= \frac{n - 2 \cdot |U(X) + X^l V(X)|}{n}, \qquad (2.2.16)$$

where we have used the fact that the distance between two linear words U(X) and V(X) is equal to the weight of the summed word U(X)+V(X).

2.3 MAXIMAL-LENGTH SEQUENCE CODES

An $(n=2^m-1, k=m)$ binary M-sequence code is generated from an m-stage linear feedback shift register by taking a binary primitive polynomial of degree m as the associated recursion polynomial. Earlier in Section 2.1, we mentioned that the minimal polynomial of a primitive element used in defining Galois field $GF(2^m)$ is a primitive polynomial of degree m and exponent $n=2^m-1$. Hence with the minimal polynomial of a primitive element of $GF(2^m)$, we can associate an M-sequence code.

The codes resulting from maximal-length linear shift register generators have several important properties.

I). Balance Property: Any M-sequence word of length $n=2^m-1$ contains 2^{m-1} 1's and $2^{m-1}-1$ 0's. That is, in each period of the sequence the number of Ones differ from the number of Zeros by at most 1.

II). Run Property: A run is defined to be a maximal string of consecutive identical symbols in the code sequence. For M-sequence word, among the runs of 1's and 0's in each period, one half of the runs of each kind are of length 1, one-quarter of each kind are of length 2, one-

eighth of length 3, and so on, as long as these fractions give integral numbers of runs. Thus, the statistical distributions are well defined and always the same, but the relative position of the runs varies from code to code.

III). Window Property: If a window of width m is slid along a M-sequence of period $n=2^m-1$, each of the 2^m-1 nonzero binary m-tuples is seen exactly once. This permits monitoring the code generation process to ensure proper operation.

IV). Generic Property: Modulo-2 addition of a linear M-sequence codeword with a shifted replica of itself results in another replica with a phase shift different from either replica.

V). Weight Distribution Property: For any maximal-length sequence code of length $n=2^m-1$, the weight of the nonzero codeword V(X) is constantly given by

$$|V(X)| = 2^{m-1} = \frac{n+1}{2}. \qquad (2.3.1)$$

Thus the weight distribution of M-sequence code is as the following:

$$\begin{aligned} N(0) &= 1, \\ N(\tfrac{n+1}{2}) &= n. \end{aligned} \qquad (2.3.2)$$

VI). Autocorrelation Property: From Properties IV) and V) together with Equ.(2.2.16), the auto-correlation function of M-sequence code of length $n=2^m-1$ is given by

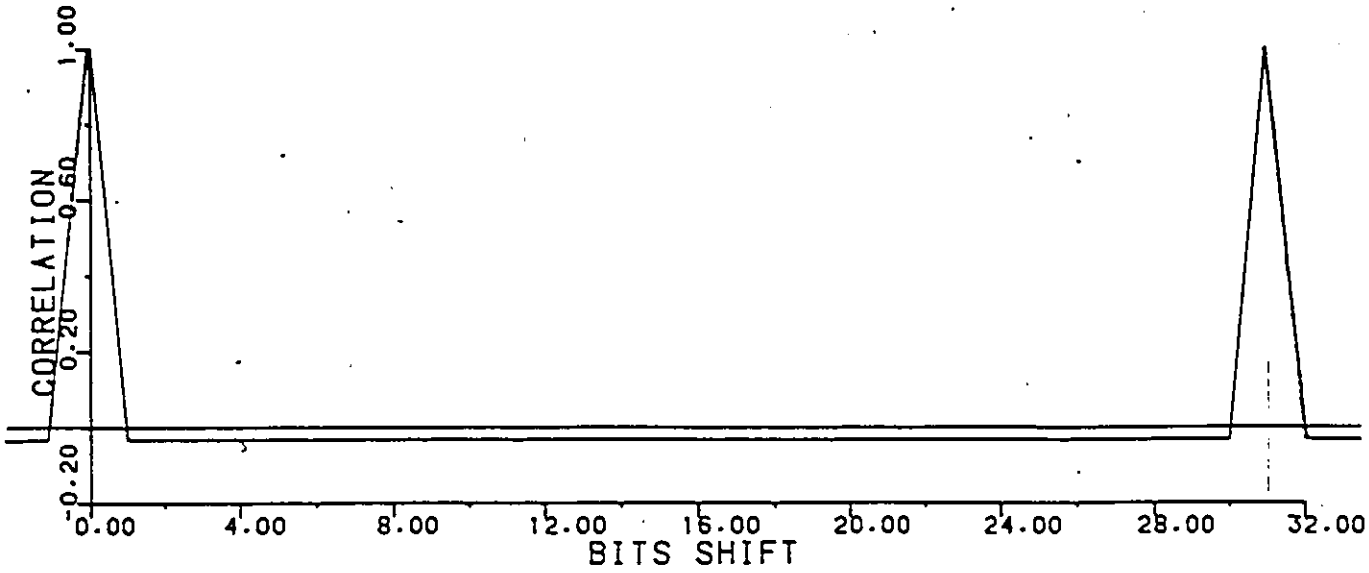
$$\theta(l) = \begin{cases} 1 \ , \ l = 0; \\ -1/n, \ 1 \le l < 2^m-1. \end{cases} \qquad (2.3.3)$$

This is the best possible auto-correlation function of any binary sequence of length $n=2^m-1$, in the sense of minimizing $\max_{0<l<n} \theta(l)$. Figure 2.3.1(a) shows such an idealized auto-correlation function.

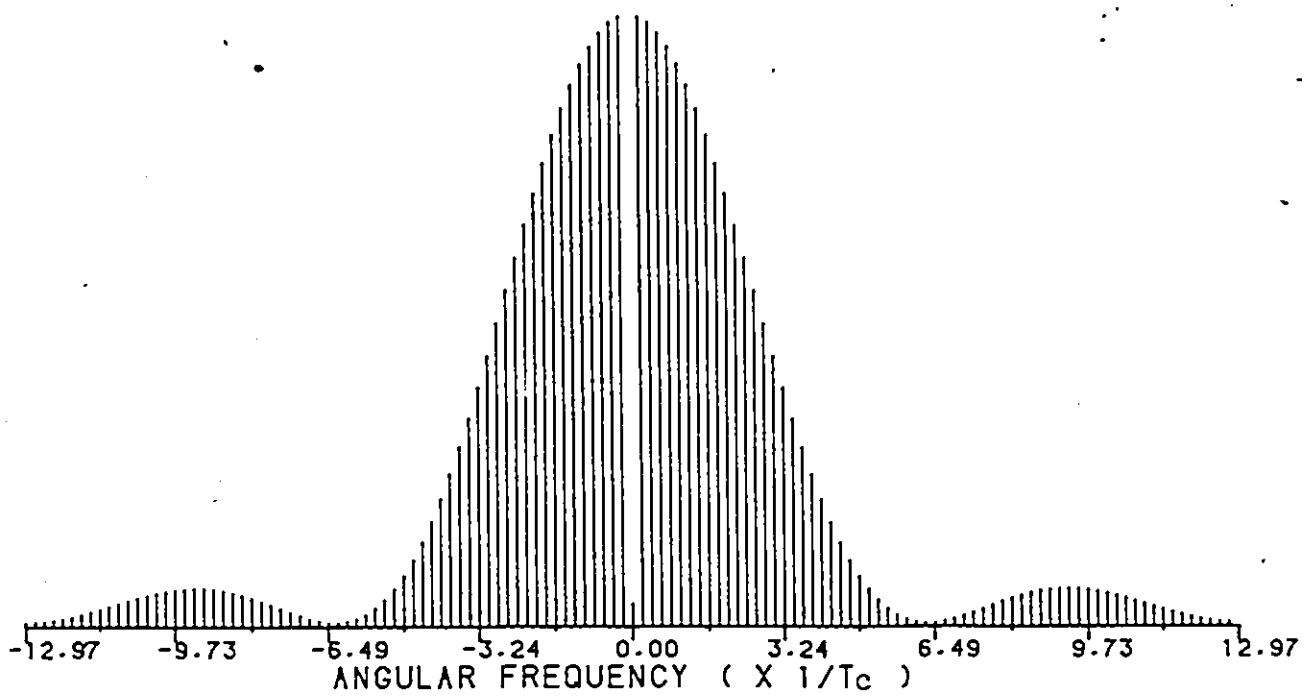
VII). Power Spectrum Property: Associated with correlation function is power spectrum. This entity can be arrived at by making use of the well known Fourier transform relationship between the correlation function and the power spectrum. For a binary waveform corresponding to an M-sequence code of length n, the power spectrum is known [10] to be

$$S(\omega) = \left(\frac{n+1}{n}\right)^2 \left[\frac{\sin\frac{\omega T_c}{2}}{\frac{\omega T_c}{2}}\right]^2 \sum_{\substack{i=-\infty \\ i \neq 0}}^{\infty} \delta\left(\omega - \frac{2\pi i}{nT_c}\right) + \frac{1}{n^2}\delta(\omega), \qquad (2.3.4)$$

where T_c is the time duration of one digit of the binary waveform, and $\delta(.)$ is the Dirac delta function, or impulse function.



(a)



(b)

Figure 2.3.1: Autocorrelation function and power spectrum for 31-bit M-sequence code. (a) Autocorrelation; (b) Power spectrum.

Figure 2.3.1(b) gives the power spectrum corresponding to the auto-correlation function of Figure 2.3.1(a). From this diagram together with Equ.(2.3.4), a number of features of the spectrum have to be noted. First, it is a line spectrum with frequencies at multiples of the fundamental frequency $1/nT_c$. Second, there is a scale factor inversely proportional to the length of the code. Thus, if the length of the code is doubled, the lines in the spectrum become twice as dense, but the power in each is reduced by a factor of 2. This is because the binary waveform is a constant amplitude square wave and hence has constant power. Third, the envelope of the spectrum, a (sinX/X) function, is independent of the length of the waveform code, but is determined solely by the digit duration of the waveform. Finally, the DC term shows a power of $1/n^2$. This results because of the odd number of digit periods in the code sequence, of which $(n+1)/2$ are of one polarity and $(n-1)/2$ are of the other polarity.

VIII). Decimation Property: If $V_1(X)$ is an M-sequence word of length $n=2^m-1$ and λ is a positive integer, then $V_\lambda(X)$ formed by taking every (λ mod n)th bit of $V_1(X)$ is said to be a decimation by λ of $V_1(X)$. Assume that $V_1(X)$ is generated by the polynomial $M_1(X)$ and is not identically zero, then the codeword $V_\lambda(X)$ has period $n/GCD(n,\lambda)$, and is generated by the polynomial $M_\lambda(X)$ whose roots are the λth powers of the roots of $M_1(X)$.

Clearly, code V_λ has length n if and only if GCD(n,λ)=1. In this case, the decimation is called a proper decimation, and the code V_λ is an M-sequence code of length n generated by the primitive polynomial $M_\lambda(X)$.

As an example, consider the irreducible polynomials of Figure 2.1.2. If the primitive polynomial $M_1(X)=1+X+X^6$ is used to generate the M-sequence code V_1, then the decimation by 3 of V_1 is the code V_3 which can equivalently be generated from $M_3(X)=1+X+X^2+X^4+X^6$. Similarly, the decimation by 5 of V_1 is the code V_5 which can be generated by $M_5(X)=1+X+X^2+X^5+X^6$, and the code V_7 decimated by 7 of V_1 can be generated from $M_7(X)=1+X^3+X^6$, and so on. The period of V_1 is 63; that of V_3 is 63/GCD(63,3)=21, and thus is not an M-sequence code; V_5 has period 63 and is an M-sequence code; V_7 has period 63/GCD(63,7)=9, and is not an M-sequence code; and so on.

Chapter III

CORRELATION OF CERTAIN COMPOSITE CODES

In this chapter, we will consider large sets of periodic code sequences which have good periodic correlation as measured by the peak periodic correlation parameters θ_a and θ_c. These are the Gold codes and the generalized composite codes. Both of them are derived from M-sequence codes and both can be generated by linear feedback shift registers of relatively short length compared with the code period.

3.1 CORRELATION OF GOLD COMPOSITE CODES

The family of Gold codes is constructed from the bit-by-bit modulo-2 addition of a preferred pair of M-sequence codes. Since the preferred M-sequence codes are of the same length, the Gold codes generated are the same length as the two base codes, but are nonmaximal. A set of Gold codes of period $n=2^m-1$ consists of $n+2$ codes for which the cross-correlation and the out-of-phase auto-correlation between any codewords of the set are well-bounded by $\theta(m)=(2^{\lfloor (m+2)/2 \rfloor}+1)/n$. Due to this bounded correlation, the Gold codes are attractive for applications in which a number of code-division-multiplexed signals are to be used. The

same guarantee of bounded correlation is impossible for M-sequence codes of the same length.

The studies of Gold codes were originated from the work [8], [9] of Gold in 1967 and 1968 respectively. At that time, Gold showed that the composite codes made up of the preferred pair of M-sequence codes have the preferred correlation values. Gold also proved that over $GF(2^m)$, if $m \not\equiv 0 \bmod 4$ and $\lambda = 2^{\lfloor (m+2)/2 \rfloor}+1$, the minimal polynomials $M_1(X)$ and $M_\lambda(X)$ of the primitive elements α and α^λ are the preferred pair. At about the same time, Kasami [13] developed the weight distribution of BCH codes and found that the nonzero words generated by $M_1(X)M_\lambda(X)$ have weights 2^{m-1} and $2^{m-1\pm 2^{\lfloor m/2 \rfloor}}$ when $m \not\equiv 0 \bmod 4$, while if $m \equiv 0 \bmod 4$, weights $2^{m-1\pm 2^{\lfloor m/2 \rfloor}-1}$ also occurred. Following these pioneering work of Gold and Kasami, Sarwate and Pursley [25] translated the above weight spectra of BCH codes into the equivalent correlation spectra of Gold codes. By arranging some suitable decimation values, they also derived several subsets or supersets of Gold codes, which have the lower or the same correlation bounds.

Here we will base on the fundamental coding theory to derive the equivalent generator polynomials (or recursion polynomials) of Gold codes. These polynomials are related with two suitable primitive polynomials. To make the code

generated to have maximum possible minimum weight or distance, and hence have a minimum peak correlation value, the two based primitive polynomials have to be suitably chosen. To do this, we rely upon BCH code's generator polynomial structure to estimate Gold code's minimum distance. Following computer searching program, we find that this minimum distance occurs when the two based primitive polynomials are the preferred polynomials, a result just as proposed by Gold and Kasami.

3.1.1 THE GENERATOR STRUCTURES OF GOLD CODES

Let $h_i(X)$ and $h_j(X)$ represent two irreducible factors of the polynomial $1+X^n$, each with exponent n. If these two factors are taken as the recursion polynomials of two linear cyclic codes, then corresponding to $h_i(X)$, the generator polynomial is

$$g_i(X) = \frac{1 + X^n}{h_i(X)}, \qquad (3.1.1)$$

and corresponding to $h_j(X)$, the generator polynomial is

$$g_j(X) = \frac{1 + X^n}{h_j(X)}. \qquad (3.1.2)$$

Therefore, the codewords generated will respectively be

$$V_i(X) = \frac{1 + X^n}{h_i(X)} I_i(X) \tag{3.1.3}$$

and

$$V_j(X) = \frac{1 + X^n}{h_j(X)} I_j(X), \tag{3.1.4}$$

where $I_i(X)$ and $I_j(X)$ are the respective information polynomials, and the multiplications are understood to be modulo $1+X^n$ over GF(2). Combining the above two code polynomials with modulo-2 addition rule, we get

$$\begin{aligned} V_i(X)+V_j(X) &= \left\{\frac{1 + X^n}{h_i(X)} I_i(X) + \frac{1 + X^n}{h_j(X)} I_j(X)\right\} \\ &= \frac{1 + X^n}{h_i(X)h_j(X)} \left\{I_i(X)h_j(X) + I_j(X)h_i(X)\right\} \\ &= g_{ij}(X) \left\{ I_i(X)h_j(X) + I_j(X)h_i(X)\right\}, \end{aligned} \tag{3.1.5}$$

where

$$g_{ij}(X) = \frac{1 + X^n}{h_i(X)h_j(X)} = \frac{1 + X^n}{h_{ij}(X)}. \tag{3.1.6}$$

From Eqs.(3.1.5) and (3.1.6), we see that $V_i(X)+V_j(X)$ is a multiple of the polynomial $g_{ij}(X)$ and hence is a word in a certain cyclic code which has $g_{ij}(X)$ as generator

polynomial. Therefore, if we let V_{ij} denote the composite code made up of V_i and V_j, then $g_{ij}(X)$ is obviously the generator polynomial of code V_{ij}, or equivalently $h_{ij}(X)=h_i(X)h_j(X)$ is the recursion polynomial of V_{ij}. Regarding to the above discussion, we conclude the following:

THEOREM 3.1.1:

Let $h_i(X)$ and $h_j(X)$ denote two irreducible binary polynomials, each with exponent n. If V_i and V_j are the two codes of period n generated with $h_i(X)$ and $h_j(X)$ as the respective recursion polynomials, then the composite code V_i+V_j constructed from the modulo-2 combination of V_i and V_j is equivalent to the code V_{ij} generated from the recursion polynomial $h_{ij}(X)=h_i(X)h_j(X)$.

Figure 3.1.1 shows the commonly adopted configuration with which the composite code V_{ij} are generated. Note that the feedback function f_i (f_j) of the shift register generators SRGi (SRGj) is a series of switches which are open or closed depending on the coefficients of $h_i(X)$ ($h_j(X)$) be 0 or 1. Also note that in the above discussion, we haven't made the usual restriction that $V_i(X)$ and $V_j(X)$ are nonzero codewords [32].

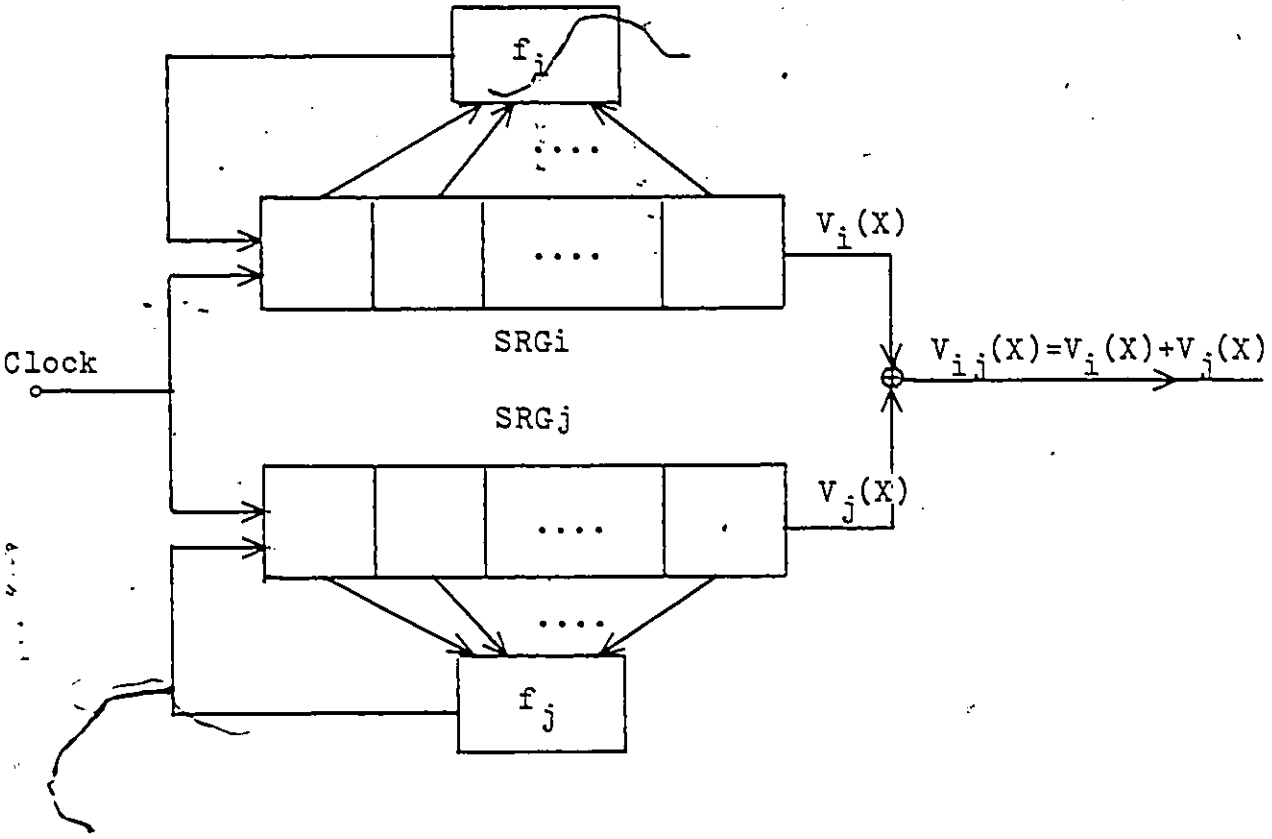


Figure 3.1.1: Typical Gold code generator configuration.

Now suppose that $h_i(X)$ and $h_j(X)$ are the two primitive polynomials of degree m that generate the M-sequence codes V_i and V_j, respectively, of period $n=2^m-1$. If $V_{ij}(X)$ denotes a nonzero codeword generated from the configuration of Figure 3.1.1, then we get either

$$V_{ij}(X) = V_i(X) \qquad (3.1.7)$$

or

$$V_{ij}(X) = V_j(X) \qquad (3.1.8)$$

or

$$V_{ij}(X) = V_i(X) + V_j(X). \qquad (3.1.9)$$

Among the above three expressions, the first two correspond to the case where one of the two base words is zero, while the last one corresponds to the case where both base words are nonzeros. The shift and add property (Generic Property) of M-sequence codes tells us that any M-sequence word modulo-2 added to a phase-shifted replica of itself produces a different phase-shifted word. Here the same operation is performed, but with the base codewords taken from two different M-sequence codes. Every change in the relative phase displacement between the two base words causes a new code to be generated. To show this advantage, consider the following example:

Given two 5-stage shift register code generators, we choose a pair of primitive binary polynomials of degree 5, e.g., $h_i(X)=1+X^2+X^5$ and $h_j(X)=1+X+X^2+X^4+X^5$, and connect the circuit into Gold configuration, as shown in Figure 3.1.2.

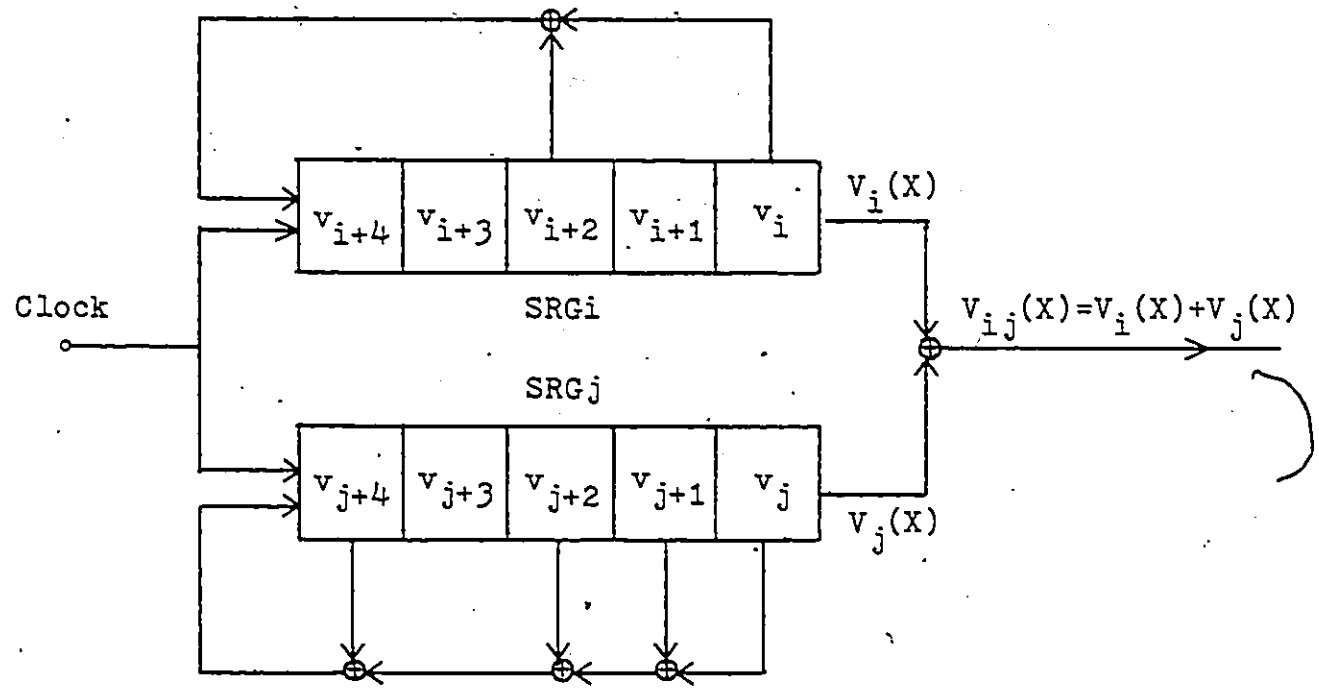


Figure 3.1.2: Illustration of Gold code generator.

By setting various initial states in the shift registers, different output codes will be generated. For the all-ones vector set in both shift registers as the initial state, the output is

$$V_i(X) \longrightarrow 11111000110111010100001001011 00$$
$$+\ V_j(X) \longrightarrow 11111010001001010110000111001 10$$
$$V_{ij}(X) \longrightarrow 00000010111110000010001110010 10$$

Now if (0,1,1,1,1) is instead set as the initial state of the shift register generator SRGj, then $V_j(X)$ has 1-bit shift from the above case, and we have

$$V_i(X) \longrightarrow 11111000110111010100001001011 00$$
$$+\ V_j(X) \longrightarrow 11110100010010101100001110011 01$$
$$V_{ij}(X) \longrightarrow 00001100100101111000000111000 01$$

Similarly, when (0,0,0,1,0) is set as the initial state of SRGj, $V_j(X)$ has 5-bits offset to the original case, and

$$V_i(X) \longrightarrow 11111000110111010100001001011 00$$
$$+\ V_j(X) \longrightarrow 01000100101011000011100110111 11$$
$$V_{ij}(X) \longrightarrow 10111100011100010111101111100 11$$

Any shift in initial condition for $V_j(X)$ from zero to 30 bits can be used (a 31-bit shift is the same as the zero shift); and for each relative shift the resulted composite code is different and is nonmaximal. Thus, from the code generator of Figure 3.1.2, 31 nonmaximal-length codes and 2 maximal-length codes are available. Extending this demonstration, we can show that any two-register Gold code generator of m-stages can generate $n=2^m-1$ nonmaximal-length codes plus the two maximal-length base codes, each of them has length of n. From the above discussion, it follows that

$V_{ij}(X)$ is some phase of some codes in the set V_{ij} defined by

$$V_{ij} = \{V_i(X),\ V_j(X),\ V_i(X)+V_j(X),\ V_i(X)+XV_j(X),$$
$$V_i(X)+X^2V_j(X),\ \ldots,\ V_i(X)+X^{n-1}V_j(X)\}, \tag{3.1.10}$$

and therefore we have the following theorem.

THEOREM 3.1.2:

Let V_i and V_j denote two M-sequence codes of length $n=2^m-1$ generated by the primitive binary polynomials $h_i(X)$ and $h_j(X)$ respectively. Then every code in the set V_{ij} defined in Equ.(3.1.10) can be generated by the recursion polynomial $h_{ij}(X)=h_i(X)h_j(X)$ or generated by adding together (term by term, modulo-2) the outputs of the shift registers corresponding to $h_i(X)$ and $h_j(X)$. The set V_{ij} contains $n+2=2^m+1$ members, each having period of n. Among these, the n codes are nonmaximal while the remaining 2 codes are the maximal-length codes V_i and V_j themselves.

3.1.2 CORRELATION SPECTRA OF GOLD CODES

In addition to their advantage in generating large numbers of codes, the Gold codes are chosen so that over a set of codes available from a given generator the cross-correlation between the codes is uniform and bounded. To see this, let us consider the following.

Suppose U and V are two distinct codes belonging to the set V_{ij} of Equ.(3.1.10). Since both these codes are generated by $g_{ij}(X)$, so is the code W=U+V. If W(X) is of the form (3.1.7) or (3.1.8), we get from the weight distribution property of M-sequence codes that

$$|W(X)| = (n+1)/2; \qquad (3.1.11)$$

while if W(X) is of the form (3.1.9) then

$$|W(X)| = |U(X)+V(X)| = |V_i(X)+V_j(X)|. \qquad (3.1.12)$$

Consequently, the correlation between codes U and V, $\theta(U,V) = (n-2.|U(X)+V(X)|)/n$, is either equal to $-1/n$ or equal to $(n-2.|V_i(X)+V_j(X)|)/n = \theta(V_i,V_j)$. A similar analysis can be carried out for the auto-correlation function $\theta(W,W)$ and shows that the set of values taken on by the correlation functions for codes in V_{ij} is just the set of values taken

on by the correlation functions for V_i and V_j. More importantly, the peak correlation parameters θ_a and θ_c for V_{ij} satisfy

$$\theta_a = \theta_c = \max_{i,j} \{ |\theta(V_i, V_j)| \}. \qquad (3.1.13)$$

In other words, given a pair of M-sequence codes V_i and V_j with peak periodic cross-correlation magnitude Λ, we can construct a set of n+2 codes with peak periodic cross-correlation magnitude and peak out-of-phase periodic auto-correlation magnitude equal to Λ. Theorem 3.1.3 below summarize this result.

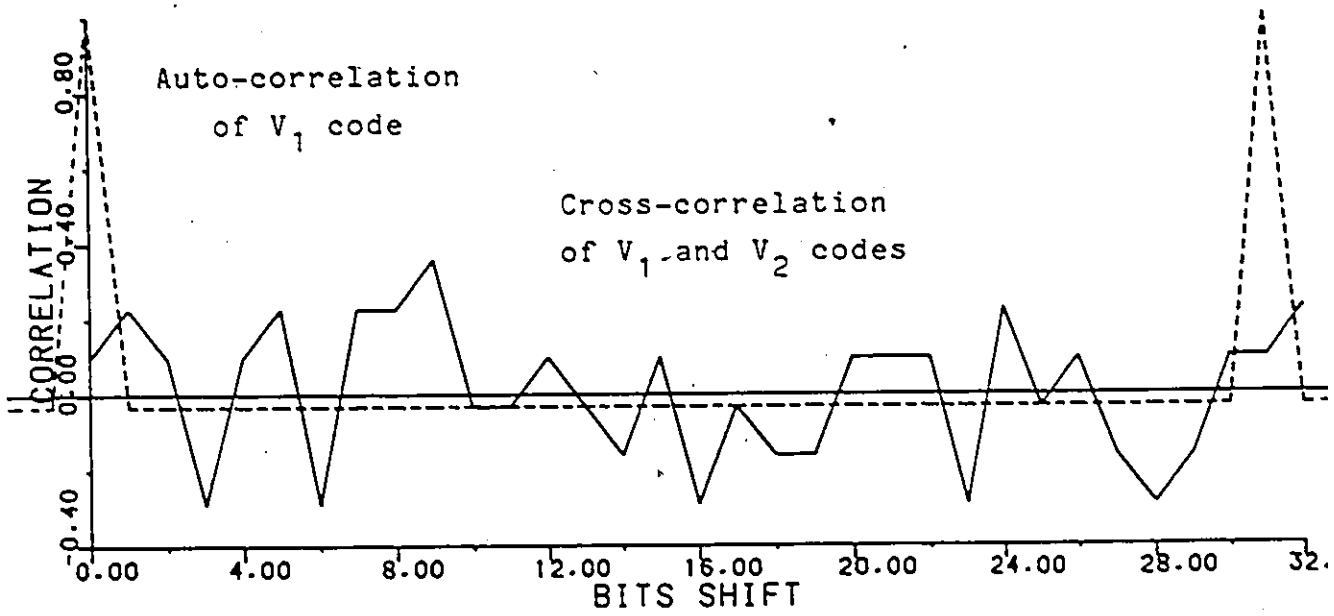
THEOREM 3.1.3:

Let V_i and V_j denote two M-sequence codes of length $n=2^m-1$. If the cross-correlation magnitude $|\theta(V_i,V_j)|$ between V_i and V_j is bounded by Λ, then the composite code V_{ij} constructed according to (3.1.10) has peak correlation magnitudes also bounded by Λ.

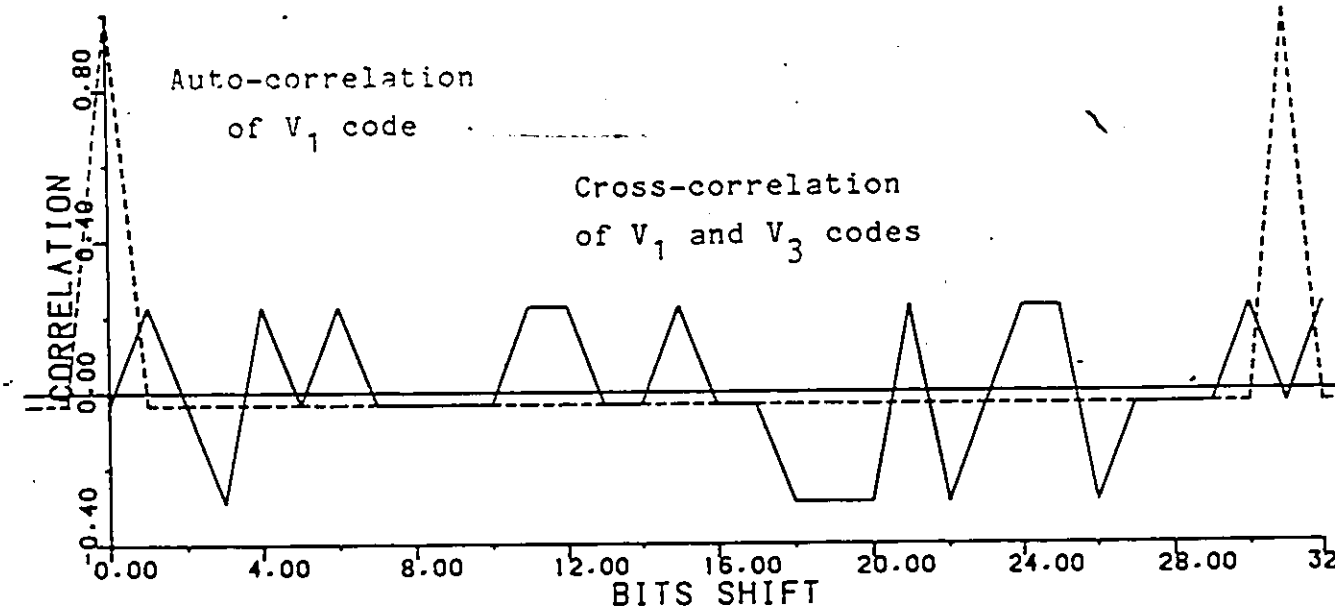
Since we always require small peak correlation magnitude for practical applications, the codes V_i and V_j have to be carefully chosen to ensure the peak correlation magnitude be as small as possible. The question of determining Gold codes' correlation properties thus reduces to that of selecting pairs of M-sequence codes with smallest peak cross-correlation magnitudes.

Unlike the auto-correlation function shown in Figure 2.3.1(a), the cross-correlation function between any two M-sequence codes is always randomly distributed and always multi-valued. Let V_i and V_j denote two M-sequence codes of length $n=2^m-1$. Then it can shown that $\theta(V_i,V_j)(l)$ always takes the form of k/n, where k is an odd integer. In fact Helleseth [12] proves that for all l, $\theta(V_i,V_j)(l)+1/n$ is a multiple of $8/n$, except when V_i and V_j are generated by reciprocal polynomials, in which case $\theta(V_i,V_j)(l)+1/n$ is a multiple of $4/n$.

Figures 3.1.3(a) and 3.1.3(b) give illustrations of cross-correlation as compared with auto-correlation for M-sequence codes of length 31. These codes are generated by $h_1(X)=1+X^2+X^5$, $h_2(X)=h_1^*(X)=1+X^3+X^5$, and $h_3(X)=1+X+X^2+X^3+X^5$, respectively. The auto-correlation curve of the V_1 code shows an in-phase correlation value of 1.0 and an out-of-phase value of -1/31. The difference between these two values is 32/31. This is sometimes called "Index of Discrimination (ID)" --- a term used for determining the receiver's ability to recognize the proper point of code synchronization. Obviously, the lower the out-of-phase peak correlation value, the higher the ID value, and the better the code. With codes V_1 and V_2 crosscorrelated, as shown in Figure 3.1.3(a), the peak value is 11/31, which gives an index of discrimination of 20/31, or 35% less than the auto-



(a)



(b)

Figure 3.1.3: Comparative correlation values for 31-bit M-sequence codes. (a) V_1 and V_2 codes; (b) V_1 and V_3 codes.

correlation value. Cross-correlation of V_1 and V_3 codes in Figure 3.1.3(b) is lower than that of the above reciprocal codes, but is still such that the peak cross-correlation value is 7/31, a value that occurs at 10 different shift positions.

For many applications it is not necessary to know the value $\theta(V_i,V_j)(l)$ for each l. For instance it may be sufficient to have tight bounds on $|\theta(V_i,V_j)(l)|$ or to know the set of values taken on by $\theta(V_i,V_j)(l)$. Frequently, we do not need to know more than the set of correlation values together with the number of integers l $(0 \leq l < n)$ for which $\theta(V_i,V_j)(l)=c$ for each c in this set. This is referred to as the spectrum of the correlation function $\theta(V_i,V_j)$ or as the correlation spectrum for the pair of codes $[V_i,V_j]$. In many cases the correlation spectrum is much easier to evaluate than the correlation function if judicious use is made of analytical results such as those stated below. The spectrum is of course much easier to tabulate than the function.

What do such correlation spectra look like? First notice that $\theta(V_i,V_j)$ takes on positive as well as negative values. According to the weight distribution property of M-sequence code, an auto-correlation spectrum is two valued:

1	occurs	1	time
-1/n	occurs	n-1	times.

For cross-correlation spectrum, Golomb [11] has observed that if V_i and V_j are generated by different primitive polynomials, then $\theta(V_i,V_j)$ takes on at least three values. Theorem 3.1.4 below exhibits specific decimations which produce three-valued cross-correlation spectra except when m is a power of 2. This result is a composite one; various parts of it were proved by Gold [8], Kasami [13], and MacWilliams [17].

THEOREM 3.1.4:

Let V_i and V_j denote M-sequence codes of period $n=2^m-1$. If V_j is the equivalent decimation of V_i by λ, where either $\lambda=2^k+1$ or $\lambda=2^{2k}-2^k+1$, and if $e=GCD(m,k)$ is such that m/e is odd, then the spectrum of $\theta(V_i,V_j)$ is three valued and

$[-1+2^{(m+e)/2}]/n$	occurs	$2^{m-e-1}+2^{(m-e-2)/2}$ times,
$-1/n$	occurs	$2^m-2^{m-e}-1$ times,
$[-1-2^{(m+e)/2}]/n$	occurs	$2^{m-e-1}-2^{(m-e-2)/2}$ times.

Notice that if e is large, $\theta(V_i,V_j)$ takes on large values but only very few times while if e is small, $\theta(V_i,V_j)$ takes on small values more frequently. In most instances, small values of e are desirable. If we wish to have e=1 then

clearly m must be odd in order that m/e be odd. When m is odd, we can take k=1 or k=2 (and possibly other values of k as well), and obtain that $\theta(V_1,V_3)$, $\theta(V_1,V_5)$ and $\theta(V_1,V_{13})$ all have the three-valued spectrum given in Theorem 3.1.4 (with e=1). Suppose next that m=2 mod 4. Then, m/e is odd if e is even and a divisor of m. Letting k=2, we obtain that $\theta(V_1,V_5)$ and $\theta(V_1,V_{13})$ both have the three-valued spectrum given in Theorem 3.1.4 (with e=2). Summarizing the above, we conclude that for $m \not\equiv 0 \bmod 4$, there exist pairs of M-sequence codes of length $n=2^m-1$ with three-valued cross-correlation functions,

$$\theta(V_i,V_j) = \begin{cases} (-1+2^{\lfloor (m+2)/2 \rfloor})/n \\ -1/n \\ (-1-2^{\lfloor (m+2)/2 \rfloor})/n \end{cases}, \qquad (3.1.14)$$

where $\lfloor x \rfloor$ denotes the integer part of the real number x.

As for the case of m being multiple of 4, i.e., $m = 0 \bmod 4$, the recent survey by Sarwate and Pursley [25] has pointed out that there exist pairs of M-sequence codes which give four-valued spectra which are considerably better than the three-valued spectra. The following theorem is due to this result.

THEOREM 3.1.5:

Let V_i and V_j denote M-sequence codes of length $n=2^m-1$, where $m = 0 \bmod 4$. If V_j is the equivalent decimation of V_i by $\lambda=2^{(m+2)/2}-1$, then $\theta(V_i,V_j)$ has a four-valued spectrum and

$$\begin{array}{llll} [-1+2^{(m+2)/2}]/n & \text{occurs} & [2^{m-1}-2^{(m-2)/2}]/3 & \text{times,} \\ [-1+2^{m/2}]/n & \text{occurs} & 2^{m/2} & \text{times,} \\ -1/n & \text{occurs} & 2^{m-1}-2^{(m-2)/2}-1 & \text{times,} \\ [-1-2^{m/2}]/n & \text{occurs} & [2^m-2^{m/2}]/3 & \text{times.} \end{array}$$

A cross-correlation function taking on the above mentioned three-values or four-values is called a preferred three-valued or four-valued cross-correlation function and the corresponding pair of M-sequence codes (polynomials) is called a preferred pair of M-sequence codes (polynomials).

From the above discussions together with Theorem 3.1.3 and Equ.(3.1.13), we know that, for V_i and V_j be any of preferred pairs of M-sequence codes, the composite code V_{ij} constructed according to (3.1.10) has peak correlation parameters

$$\theta_a = \theta_c = \begin{cases} (2^{\lfloor (m+2)/2 \rfloor}+1)/n, & \text{for } m \neq 0 \bmod 4; \quad (3.1.15a) \\ (2^{(m+2)/2}-1)/n, & \text{for } m=0 \bmod 4. \quad (3.1.15b) \end{cases}$$

In this case, the cross-correlation functions for codes belonging to V_{ij} take on the preferred values only. In particular, according to Theorem 3.1.4, $[V_i, V_{\lambda i}]$ with $\lambda = 2^{\lfloor (m+2)/2 \rfloor + 1}$ is a preferred pair of M-sequence codes whenever $m \not\equiv 0 \bmod 4$. Consequently, V_{ij} formed by V_i and $V_j = V_{\lambda i}$ has peak correlation parameters be the value of (3.1.15a), and the correlation functions for codes in V_{ij} take on the preferred three values. This is the one originally devised by Gold [8], [9] in the late 1960's and have been widely adopted as the unique definition of Gold codes. While this has mnemonic value (both the decimation and the correlation bound have the term $2^{\lfloor (m+2)/2 \rfloor + 1}$), however, we would also like to extend Gold codes to the case of $m \equiv 0 \bmod 4$. With such an extension, the Gold code V_G then is, in case of $m \not\equiv 0 \bmod 4$, constructed from the M-sequence codes pair $[V_i, V_{\lambda_1 i}]$, $\lambda_1 = 2^{\lfloor (m+2)/2 \rfloor + 1}$; while in case of $m \equiv 0 \bmod 4$ it is modulo-2 summed from the M-sequence codes pair $[V_i, V_{\lambda_2 i}]$, $\lambda_2 = 2^{(m+2)/2} - 1$. In any respect, V_G should be called a set of Gold codes whenever $[V_i, V_{\lambda i}]$ is any preferred pair of M-sequence codes. We summarize the above discussion as follows.

THEOREM 3.1.6:

Let V_i and V_j denote two preferred M-sequence codes of length $n=2^m-1$ generated respectively by the primitive binary polynomials $h_i(X)$ and $h_j(X)$, where the roots of $h_j(X)$ are the λ-th powers of those of $h_i(X)$, $\lambda = 2^{\lfloor (m+2)/2 \rfloor}+1$ for $m \neq 0 \bmod 4$, and $\lambda = 2^{(m+2)/2}-1$ for $m = 0 \bmod 4$. Then the set V_G defined according to Equ.(3.1.10) is a set of Gold codes. For $U,V \in V_G$, the cross-correlation and the out-of-phase auto-correlation take the preferred three or four values only, i.e.,

$$\theta(U,V) = \begin{cases} (2^{\lfloor (m+2)/2 \rfloor}-1)/n \\ -1/n \\ (-2^{\lfloor (m+2)/2 \rfloor}-1)/n \end{cases}, \quad m \neq 0 \bmod 4. \qquad (3.1.16a)$$

or

$$\theta(U,V) = \begin{cases} (2^{(m+2)/2}-1)/n \\ (2^{m/2}-1)/n \\ -1/n \\ (-2^{m/2}-1)/n \end{cases}, \quad m = 0 \bmod 4. \qquad (3.1.16b)$$

Examining Theorem 3.1.6, we see that the preferred three-valued correlation magnitudes are bounded by

$$|\theta(U,V)| \leq \frac{2^{(m+1)/2}+1}{n}, \quad m \text{ odd}; \tag{3.1.17a}$$

or

$$|\theta(U,V)| \leq \frac{2^{(m+2)/2}+1}{n}, \quad m=2 \bmod 4; \tag{3.1.17b}$$

while the preferred four-valued correlation magnitudes are bounded by

$$|\theta(U,V)| \leq \frac{2^{(m+2)/2}-1}{n}, \quad m=0 \bmod 4. \tag{3.1.17c}$$

For sufficiently large n, the bound in (3.1.17a) tends to $\sqrt{2}/\sqrt{n}$ and the bounds in (3.1.17b) and (3.1.17c) tend to $2/\sqrt{n}$.

3.1.3 POLYNOMIAL PAIRS FOR CONSTRUCTING GOLD CODES

In the last section we mentioned that the preferred M-sequence pair $[V_i, V_j]$ for constructing a set of Gold codes V_G of length $n=2^m-1$ occurred when $\alpha^i \in GF(2^m)$ and $\alpha^j=\alpha^{\lambda i} \bmod \alpha^n \in GF(2^m)$ are respectively roots of the primitive polynomials $h_i(X)$ and $h_j(X)$, where $\lambda = 2^{\lfloor (m+2)/2 \rfloor}+1$ for $m \neq 0 \bmod 4$, and $\lambda = 2^{(m+2)/2}-1$ for $m = 0 \bmod 4$. In this section, as an alternative check, we will reexamine these criteria from Gold codes' minimum distances point of view. Also, with the aid of practical examples, we will show out how those preferred polynomial pairs can be derived. All of these efforts will strongly rely on the application of finite field theory developed earlier in Section 2.1.

According to the definition of correlation function, the correlation magnitude $\theta(V_i, V_j)$ is related with the Hamming distance $dis(V_i, V_j)$ by

$$\theta(V_i, V_j) = \frac{n-2\,dis(V_i, V_j)}{n} . \qquad (3.1.18)$$

Obviously, for a specified code length n, $\theta(V_i, V_j)$ can be as small as possible only if $dis(V_i, V_j)$ is large enough.

Since Gold codes are actually subclasses of BCH codes, they possess the same generator polynomial structures.

That is, if the generator polynomial $g(X)$ contains d_o-1 consecutive power roots, then the resulted cyclic code has a true minimum distance $d \geq d_o$. In other words, if α is a primitive element of $GF(2^m)$ and $g(X)$ has α^j, α^{j+1}, α^{j+2}, ..., α^{j+d_o-2} as roots but not α^{j-1} or α^{j+d_o-1}, then the cyclic code, of length $n=2^m-1$, generated by this $g(X)$ is assured to have a minimum distance $d \geq d_o$. In this sense, d_o is sometimes called design distance or BCH bound.

Regarding to the above fact, let's consider the following. Suppose polynomials $M_{a_1}(X)$, $M_{a_2}(X)$, ..., $M_{a_k}(X)$ are the possible irreducible factors of $1+X^n=1+X^{2^m-1}$, i.e.,

$$1+X^n = M_{a_1}(X)M_{a_2}(X)\cdots M_{a_k}(X). \qquad (3.1.19)$$

Also, among these factors, suppose $M_{a_i}(X)$ and $M_{a_j}(X)$ are the two polynomials of exponent n, then with respect to (3.1.6), if we let $h_i(X)=M_{a_i}(X)$ and $h_j(X)=M_{a_j}(X)$, the corresponding generator polynomial will be

$$g_{ij}(X) = M_{a_1}(X)M_{a_2}(X)\cdots M_{a_{i-1}}(X)M_{a_{i+1}}(X)$$
$$\cdots M_{a_{j-1}}(X)M_{a_{j+1}}(X)\cdots M_{a_k}(X). \qquad (3.1.20)$$

Among the roots of $g_{ij}(X)$ of (3.1.20) there are several sets of consecutive powers of the primitive element $\alpha \in GF(2^m)$, the one which contains the maximal consecutive members will determine the resulted code's minimum distance and hence its correlation properties.

As an example, consider the case of m=5 and n=31. For this value the irreducible polynomials table in [21] gives the entry

DEGREE 5 1 45E 3 75G 5 67H.

By following the same discussions given in Section 2.1, the cyclotomic cosets and their corresponding minimal polynomials are derived as those shown in Figure 3.1.4. Since n=31 is a prime, GCD(n,s) = 1 for all coset element s, $1 \le s \le n-1$. Thus, with reference to (2.1.3), every minimal polynomial of degree m=5 has exponent n=31. Accordingly, we have 6 possible M-sequence codes of length 31.

Cyclotomic Cosets	Minimal Polynomials
{ 1, 2, 4, 8,16}	$1+X^2+X^5$
{ 3, 6,12,24,17}	$1+X^2+X^3+X^4+X^5$
{ 5,10,20, 9,18}	$1+X+X^2+X^4+X^5$
{ 7,14,28,25,19}	$1+X+X^2+X^3+X^5$
{11,22,13,26,21}	$1+X+X^3+X^4+X^5$
{15,30,29,27,23}	$1+X^3+X^5$

Figure 3.1.4: Cyclotomic cosets and minimal polynomials for field GF(2^5) defined by $\alpha^5=1+\alpha^2$.

With reference to Figure 3.1.4, if we let

$$h_i(X) = M_1(X) = 1+X^2+X^5 \qquad (3.1.21a)$$

and

$$h_j(X) = M_3(X) = 1+X^2+X^3+X^4+X^5, \qquad (3.1.21b)$$

then the generator polynomial will be

$$g_{1,3}(X) = \frac{1+X^{31}}{M_1(X)M_3(X)}$$
$$= (1+X)(1+X+X^2+X^4+X^5)(1+X+X^2+X^3+X^5)$$
$$(1+X+X^3+X^4+X^5)(1+X^3+X^5). \qquad (3.1.21c)$$

This polynomial contains all power roots except those located in the cyclotomic cosets C_1 and C_3. In other words, $g_{1,3}(X)$ has the following roots:

$$\alpha^0 = \alpha^{31} = 1 \ ,$$
$$\alpha^5 \ , \ \alpha^{10}, \ \alpha^{20}, \ \alpha^9 \ , \ \alpha^{18},$$
$$\alpha^7 \ , \ \alpha^{14}, \ \alpha^{28}, \ \alpha^{25}, \ \alpha^{19},$$
$$\alpha^{11}, \ \alpha^{22}, \ \alpha^{13}, \ \alpha^{26}, \ \alpha^{21},$$
$$\alpha^{15}, \ \alpha^{30}, \ \alpha^{29}, \ \alpha^{27}, \ \alpha^{23}.$$

From these elements we can find that the maximal consecutive power roots of $g_{1,3}(X)$ occurs from α^{25} to $\alpha^{31}=\alpha^0=1$, hence we have $(d_o)_{1,3}-1 = 7$ or $(d_o)_{1,3} = 8$.

Now instead we take

$$h_i(X) = M_1(X) = 1+X^2+X^5 \qquad (3.1.22a)$$

and

$$h_j(X) = M_5(X) = 1+X+X^2+X^4+X^5, \qquad (3.1.22b)$$

then the generator polynomial will be

$$g_{1,5}(X) = \frac{1+X^{31}}{M_1(X)M_5(X)}$$

$$= (1+X)(1+X^2+X^3+X^4+X^5)(1+X+X^2+X^3+X^5)$$

$$(1+X+X^3+X^4+X^5)(1+X^3+X^5). \qquad (3.1.22c)$$

This polynomial has the following roots:

$$\alpha^0 = \alpha^{31} = 1 \ ,$$
$$\alpha^3 \ , \ \alpha^6 \ , \ \alpha^{12}, \ \alpha^{24}, \ \alpha^{17},$$
$$\alpha^7 \ , \ \alpha^{14}, \ \alpha^{28}, \ \alpha^{25}, \ \alpha^{19},$$
$$\alpha^{11}, \ \alpha^{22}, \ \alpha^{13}, \ \alpha^{26}, \ \alpha^{21},$$
$$\alpha^{15}, \ \alpha^{30}, \ \alpha^{29}, \ \alpha^{27}, \ \alpha^{23}.$$

From these elements we find that the maximal consecutive power roots is from α^{21} to $\alpha^{31} = \alpha^0 = 1$, so we have $(d_o)_{1,5} - 1 = 11$ or $(d_o)_{1,5} = 12$.

Following the above procedures, we can derive out all consecutive power roots of $g_{ij}(X)$. Computing result shows that the largest number of consecutive power roots of $g_{ij}(X)$'s is $(d_o)_{1,5} - 1 = 11$ which occurs when $h_i(X)$ and $h_j(X)$

are respectively the primitive binary polynomials of (3.1.22a) and (3.1.22b). Consequently, the cyclic code $V_{1,5}$ of length n=31 generated by $g_{1,5}(X)$ of (3.1.22c) is assured to have a minimum distance $(d)_{1,5} \geq 12$.

The above discussion reminds us of the famous weight distribution formula of BCH codes [13], which was developed by Kasami in 1969. According to that result, if $M_1(X)$ and $M_\lambda(X)$ are minimal polynomials of elements $\alpha \in GF(2^m)$ and $\alpha^\lambda \in GF(2^m)$, where $m \not\equiv 0 \bmod 4$ and $\lambda = 2^{\lfloor (m+2)/2 \rfloor}+1$, then the nonzero words generated by $h(X)=M_1(X)M_\lambda(X)$ have weights 2^{m-1} and $2^{m-1}\pm 2^{\lfloor m/2 \rfloor}$. With our example of m=5, we have $\lambda=2^{\lfloor (5+2)/2 \rfloor}+1=9$, and hence the code $V_{1,9}$ generated by $h(X)=M_1(X)M_9(X)$ has minimum distance $d=2^{5-1}-2^{\lfloor 5/2 \rfloor}=12$. Since 5 and 9 are in the same cyclotomic coset, α^5 and α^9 have the same minimal polynomial, i.e., $M_5(X)=M_9(X)=1+X+X^2+X^4+X^5$. The result we got here is thus the same with what we had concluded in the last paragraph. Therefore, the polynomial pair $[M_1(X), M_9(X)=M_5(X)] = [1+X^2+X^5, 1+X+X^2+X^4+X^5]$ is a preferred pair which can be taken to construct a set of Gold codes of length n=31.

So far, all discussions were confined to the field $GF(2^5)$ defined with the logic of $\alpha^5=1+\alpha^2$. Now suppose we did not take α as the element for generating all nonzero elements of $GF(2^5)$, instead we take another element $\beta=\alpha^5$ as the symbol

defining the nonzero elements of another version of $GF(2^5)$. Since corresponding to α^5, the minimal polynomial is $M_5(X)=1+X+X^2+X^4+X^5$, the logic $1+\beta+\beta^2+\beta^4+\beta^5=0$ can be used to define the new version of $GF(2^5)$. Accordingly, by utilizing the relationship of $\beta=\alpha^5$, all elements of $GF_2(2^5)$ (defined with logic $\beta^5=1+\beta+\beta^2+\beta^4$) and all elements of $GF_1(2^5)$ (defined with logic $\alpha^5=1+\alpha^2$) are found having one-to-one and onto mappings. In other words, $GF_2(2^5)$ and $GF_1(2^5)$ are isomorphic to each other. Figure 3.1.5 shows such isomorphism between $GF_2(2^5)$ and $GF_1(2^5)$ defined respectively by logics $\beta^5=1+\beta+\beta^2+\beta^4$ and $\alpha^5=1+\alpha^2$.

Elements of $GF_2(2^5)$	Elements of $GF_1(2^5)$
$\{0\}$	$\{0\}$
$\{1\}$	$\{1\}$
$\{\beta, \beta^2, \beta^4, \beta^8, \beta^{16}\}$	$\{\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^{18}\}$
$\{\beta^3, \beta^6, \beta^{12}, \beta^{24}, \beta^{17}\}$	$\{\alpha^{15}, \alpha^{30}, \alpha^{29}, \alpha^{27}, \alpha^{23}\}$
$\{\beta^5, \beta^{10}, \beta^{20}, \beta^9, \beta^{18}\}$	$\{\alpha^{25}, \alpha^{19}, \alpha^7, \alpha^{14}, \alpha^{28}\}$
$\{\beta^7, \beta^{14}, \beta^{28}, \beta^{25}, \beta^{19}\}$	$\{\alpha^4, \alpha^8, \alpha^{16}, \alpha, \alpha^2\}$
$\{\beta^{11}, \beta^{22}, \beta^{13}, \beta^{26}, \beta^{21}\}$	$\{\alpha^{24}, \alpha^{17}, \alpha^3, \alpha^6, \alpha^{12}\}$
$\{\beta^{15}, \beta^{30}, \beta^{29}, \beta^{27}, \beta^{23}\}$	$\{\alpha^{13}, \alpha^{26}, \alpha^{21}, \alpha^{11}, \alpha^{22}\}$

Figure 3.1.5: The isomorphism of elements of $GF(2^5)$'s. defined respectively by logics $\beta^5=1+\beta+\beta^2+\beta^4$ and $\alpha^5=1+\alpha^2$.

Combining Figures 3.1.4 and 3.1.5, the cyclotomic cosets and the minimal polynomials of β^i's are derived as shown in Figure 3.1.6. Here note that the $M_i(X)$ corresponding to β^i is not the same as the $M_i(X)$ corresponding to α^i. This is because the elements α^i and β^i are defined on two different versions of $GF(2^5)$. From Figure 3.1.6 the minimal polynomials of elements β and β^9 are found to be $1+X+X^2+X^4+X^5$ and $1+X+X^2+X^3+X^5$, respectively. These two polynomials thus form a preferred pair and can be used to construct another set of Gold codes.

Cyclotomic Cosets	Minimal Polynomials
{ 1, 2, 4, 8,16 }	$1+X+X^2+X^4+X^5$
{ 3, 6,12,24,17 }	$1+X^3+X^5$
{ 5,10,20, 9,18 }	$1+X+X^2+X^3+X^5$
{ 7,14,28,25,19 }	$1+X^2+X^5$
{11,22,13,26,21 }	$1+X^2+X^3+X^4+X^5$
{15,30,29,27,23 }	$1+X+X^3+X^4+X^5$

Figure 3.1.6: Cyclotomic cosets and minimal polynomials for field $GF(2^5)$ defined by $\beta^5=1+\beta+\beta^2+\beta^4$.

Surely, we can also use other symbols and logics to construct other versions of $GF(2^5)$. For example, we can take symbol $\gamma = \alpha^7$ and logic $\gamma^5 = 1+\gamma+\gamma^2+\gamma^3$, symbol $\delta = \alpha^{11}$ and logic $\delta^5 = 1+\delta+\delta^3+\delta^4$, symbol $\eta = \alpha^{15}$ and logic $\eta^5 = 1+\eta^3$, etc., to form versions of $GF(2^5)$. In all cases, the generated $GF(2^5)$'s are isomorphic to each other. From these possible versions of $GF(2^5)$, preferred pairs of polynomials can all be deduced. Appendix A of this thesis gives a complete list of such preferred polynomial pairs.

From the above procedures it is seen that all preferred pairs of polynomials of degree m can actually be deduced from any specific version of $GF(2^m)$. More precisely, for the field $GF(2^m)$ defined by logic $M_1(\alpha)=0$, based on the cyclotomic cosets and their corresponding minimal polynomials, we look for the cyclotomic cosets in which the coset elements i and $j=(\lambda i) \bmod n$ locate, $1 \le i,j \le n-1$ and λ is as defined before, then $M_i(X)$ and $M_j(X)$, the minimal polynomials of α^i and α^j, will be the preferred pair.

As a second example, let's consider m=6 and n=63. For this case, the minimal polynomial of $\alpha^i \in GF(2^6)$ (defined with logic $\alpha^6 = 1+\alpha$) had been derived and listed in Figure 2.1.2. Since n=63 is not a prime, not all of those irreducible polynomials have exponent n=63. The only polynomials which have exponent n=63 are those with root

powers of i, (n,i)=1. For our demonstration purpose, the entries of Figure 2.1.2 are reduced to those of Figure 3.1.7, where all polynomials are of degree m=6 and exponent n=63. Thus, with the length of n=63, we can have 6 possible M-sequence codes only.

Cyclotomic Cosets	Minimal Polynomials
{ 1, 2, 4, 8,16,32}	$1+X+X^6$
{ 5,10,20,40,17,34}	$1+X+X^2+X^5+X^6$
{11,22,44,25,50,37}	$1+X^2+X^3+X^5+X^6$
{13,26,52,41,19,38}	$1+X+X^3+X^4+X^6$
{23,46,29,58,53,43}	$1+X+X^4+X^5+X^6$
{31,62,61,59,55,47}	$1+X^5+X^6$

Figure 3.1.7: Certain cyclotomic cosets and minimal polynomials for field GF(2^6) defined by $\alpha^6=1+\alpha$.

Now, with reference to Figure 3.1.7, if logic $M_1(\alpha)=0$ is based to construct a version of GF(2^6), then, since $\lambda=2^{\lfloor(6+2)/2\rfloor}+1=17$, the preferred polynomials will be those having α and α^{17} as the respective roots. According to Figure 3.1.7, the minimal polynomial of α is $M_1(X)=1+X+X^6$, and the minimal polynomial of α^{17} is $M_{17}(X)=1+X+X^2+X^5+X^6$,

the polynomial pair $[M_1(X), M_{17}(X)=M_5(X)] = [1+X+X^6, 1+X+X^2+X^5+X^6]$ is therefore a preferred pair. This pair can be used to construct a set of Gold codes of length 63. Similarly, if logic $M_5(\alpha)=0$ is based to construct another version of $GF(2^6)$, then, since $j=(\lambda i) \bmod n = (17\times5) \bmod 63=22$, the minimal polynomials of elements α^5 and α^{22} form a preferred pair. That is, $[M_5(X), M_{22}(X)=M_{11}(X)] = [1+X+X^2+X^5+X^6, 1+X^2+X^3+X^5+X^6]$ is a preferred pair of polynomials. In the same way, if $GF(2^6)$ is defined from the logic of $M_{11}(\alpha)=0$, then, with i=11 and $j=(17\times11) \bmod 63=61$, the polynomial pair $[M_{11}(X), M_{61}(X)=M_{31}(X)] = [1+X^2+X^3+X^5+X^6, 1+X^5+X^6]$ is also a preferred pair.

Supplementary to the whole discussion on the correlation properties of Gold composite codes, in the Appendix A of this thesis we list out all conventionally defined preferred pairs of primitive polynomials which are suitable for constructing sets of Gold codes. Also, in the Appendix B the FORTRAN program on which those lists of Appendix A are based is included for reference.

3.2 CORRELATION OF GENERALIZED COMPOSITE CODES

Having investigated the correlation properties of the famous Gold codes, now let us turn to consider another important class of composite codes, which we shall call generalized composite codes. With a few similarity to the construction of Gold codes, these composite codes are combined, in a bit-by-bit modulo-2 addition basis, from any relatively prime length linear codes. In other words, if V_i is an (n_i,k_i) binary linear code for $i = 1, 2, \ldots, \mu$, where n_i's are pairwise relatively prime, then by repeating each codeword $V_i(X) \in V_i$ to a common length $n=n_1 n_2 \ldots n_\mu$, we can combine them into the composite codeword $V^{(\mu)}(X) = \sum_{i=1}^{\mu} V_i(X)\frac{1+X^n}{1+X^{n_i}} \in V^{(\mu)}$. In this section, we discuss the question of the correlation between codewords of $V^{(\mu)}$ for the case when all of the component codes V_i are M-sequence codes. Since codewords' correlation is intimately related with their Hamming weights, we will firstly examine the weight distributions of $V^{(\mu)}$. To ease the presentation, the derivations of weight distributions and some other related properties of $V^{(\mu)}$ will be based on the case of $\mu=2$. Following the linearity characteristics of linear codes, the results for $\mu=2$ can always be extended to any $\mu>2$ simply via recursive applications.

3.2.1 GENERAL PROPERTIES OF GENERALIZED COMPOSITE CODES [27]

Let F be the binary field GF(2) and F^n the vector space of all binary n-tuples. Let V_i be an (n_i, k_i) binary linear code for $i = 1, 2$, where n_1 and n_2 are relatively prime. With $n = n_1 n_2$, the function ϕ_i, $i = 1, 2$, defined by

$$\phi_i : v_i \longrightarrow (v_i, v_i, \ldots, v_i), \tag{3.2.1}$$

where $v_i \in V_i$ and is repeated n/n_i times, is an injective linear map from V_i into F^n. Let $\tilde{V}_i$ be the image of V_i under ϕ_i. Then $\tilde{V}_i$ is a linear subspace of F^n of dimension k_i. By combining $\tilde{V}_i$ in a bit-by-bit modulo-2 addition fashion, we get a new linear code $V^{(2)}$ defined by

$$V^{(2)} = \tilde{V}_1 + \tilde{V}_2. \tag{3.2.2}$$

This code has length $n = n_1 n_2$ and dimension

$$k = k_1 + k_2 - \dim(\tilde{V}_1 \cap \tilde{V}_2). \tag{3.2.3}$$

Since n_1 and n_2 are relatively prime it easily follows that

$$\dim(\tilde{V}_1 \cap \tilde{V}_2) = \begin{cases} 1, & \text{if } 1^{n_1} \in V_1 \text{ and } 1^{n_2} \in V_2, \\ 0, & \text{otherwise,} \end{cases} \tag{3.2.4}$$

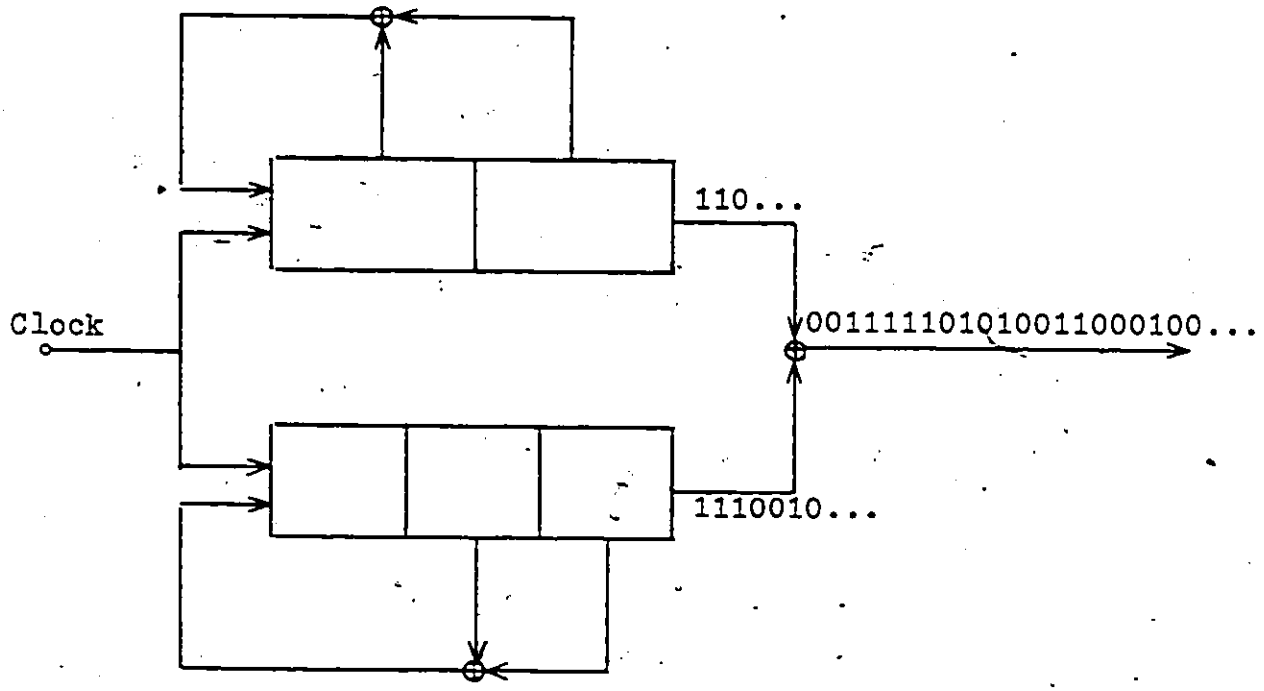
where 1^{n_i} is the all-ones word of length n_i, $i = 1, 2$. Using this fact in (3.2.3), we have the following:

THEOREM 3.2.1:

If V_i is an (n_i,k_i) binary linear code for $i = 1, 2$, and n_1 and n_2 are relatively prime, then the binary linear code $V^{(2)}$, as defined in (3.2.2), is (n,k) code with $n=n_1n_2$ and $k=k_1+k_2-\alpha$, where $\alpha=1$ if both V_1 and V_2 contain the all-ones word and $\alpha=0$ otherwise.

For case of V_i, i=1, 2, be M-sequence code, there is no all-ones codeword exists, so $\alpha=0$, and the dimension of V is $k=k_1+k_2$.

The above statements can be clarified by examining the example of Figure 3.2.1. With suitable initial states loaded on the memory cells and with the shift register feedback circuits connected like that shown in Figure 3.2.1(a) (i.e., with $h_1(X)=1+X+X^2$ and $h_2(X)=1+X+X^3$), the M-sequence codes V_1 and V_2 of periods $n_1=3$ and $n_2=7$ will respectively be generated from their shift register circuits. The corresponding code bits of these two code sequences are modulo-2 added together during each unit of clock time and results in the generalized composite code of period $n=n_1n_2=21$, as can be seen from Figure 3.2.1(b).



(a)

$\tilde{V}_1(X) \rightarrow$ 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 ...

$+ \tilde{V}_2(X) \rightarrow$ 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0

$V^{(2)}(X) \rightarrow$ 0 0 1 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 1 0 0

(b)

Figure 3.2.1: An example for generating generalized composite code.

Note that with all possible initial loadings on the above shift register code generator, the possible codewords number of V_1 is $2^{k_1}=(1+n_1)=4$, and the possible codewords number of V_2 is $2^{k_2}=(1+n_2)=8$. Moreover, none of these words is the all-ones word. Hence the possible number of codewords of the generalized composite code $V^{(2)}$ is $2^{k_1+k_2} = (1+n_1)(1+n_2) = 32$. These words are distributed among the following four cyclic classes:

1). One cyclic class is the all-zeros word. This corresponds to the case of both $V_1(X)$ and $V_2(X)$ are zero codewords;

2). One cyclic class contains $n_1=3$ cyclic codewords. This corresponds to the case of $V_1(X) \neq 0$ and $V_2(X)=0$;

3). One cyclic class contains $n_2=7$ cyclic codewords. This corresponds to the case of $V_1(X)=0$ and $V_2(X) \neq 0$;

4). One cyclic class contains $n=n_1n_2=21$ cyclic codewords. This corresponds to the case of both $V_1(X)$ and $V_2(X)$ are nonzero codewords.

It is clear that Theorem 3.2.1 can be easily extended, by a recursive application, to the case of μ (n_i,k_i) component codes V_i, where n_i's are pairwise relatively prime, for any

$\mu > 2$. With such an extension, the generalized expression of the dimensionality of $V^{(\mu)}$ will be

$$k = \{[\ldots[[k_1+k_2-\dim(\tilde{V}_1\cap\tilde{V}_2)]+k_3-\dim(\tilde{V}^{(2)}\cap\tilde{V}_3)] +\ldots]+k_\mu-\dim(\tilde{V}^{(\mu-1)}\cap\tilde{V}_\mu)\}$$

$$= \sum_{i=1}^{\mu} k_i - \sum_{j=1}^{\mu-1} \dim(\tilde{V}^{(j)} \cap \tilde{V}_{j+1}), \qquad (3.2.5)$$

where the values of $\dim(\tilde{V}^{(j)} \cap \tilde{V}_{j+1})$, $j=1, 2, \ldots, \mu-1$, are either 0 or 1, depending on whether the all-ones words are in both $\tilde{V}^{(j)}$ and $\tilde{V}_{j+1}$, such as that depicted in (3.2.4). For V_i's be M-sequence codes, the structure of maximal-length code tells us that none of the above $\tilde{V}_{j+1}$ is the all-ones code, hence the dimension of (3.2.5) reduces to

$$k = \sum_{i=1}^{\mu} k_i. \qquad (3.2.6)$$

The possible codewords number of generalized composite code made up of μ M-sequence component codes is thus $2^k = \prod_{i=1}^{\mu}(1+n_i)$. These words are distributed among the following $2^\mu = \sum_{i=0}^{\mu}\binom{\mu}{i}$ cyclic classes:

* $\binom{\mu}{0}=1$ cyclic class with the all-zeros word;
* $\binom{\mu}{1}=\mu$ cyclic classes have their codewords numbers respectively be n_i, $i=1, 2, \ldots, \mu$;
* $\binom{\mu}{2}=\mu(\mu-1)/2$ cyclic classes have their codewords numbers respectively be $n_i n_j$, $i < j \leq \mu$;
* .....
* .....

* $\binom{\mu}{\mu-1}=\mu$ cyclic classes have their codewords numbers respectively be $(\prod_{i=1}^{\mu} n_i)/n_i$, i=1, 2, ..., μ;
* $\binom{\mu}{\mu}=1$ cyclic class with $\prod_{i=1}^{\mu} n_i$ cyclic codewords.

Figure 3.2.2 shows the commonly adopted generator configuration of generalized composite code made up of μ pairwise relatively prime lengths component codes.

Just like that of Gold code, the generalized composite code constructed with the configuration of Figure 3.2.2 has also an equivalent generator polynomial. In the Theorem 3.2.2 below we discuss the case when, with respect to Equ.(3.2.2), the component codes V_i's, i=1, 2, are cyclic. Here we give the direct results only, for the formal proof the readers can consult with the work of [27].

THEOREM 3.2.2:

If V_i is cyclic with the generator polynomial $g_i(X)$, i=1, 2, then $V^{(2)}$ is cyclic with the generator polynomial

$$g^{(2)}(X) = \mathrm{GCD}\left\{\frac{g_1(X)(1+X^n)}{1+X^{n_1}}, \frac{g_2(X)(1+X^n)}{1+X^{n_2}}\right\} . \qquad (3.2.7)$$

The recursive applicaton of Theorem 3.2.2 to the case of μ component codes results in the following:

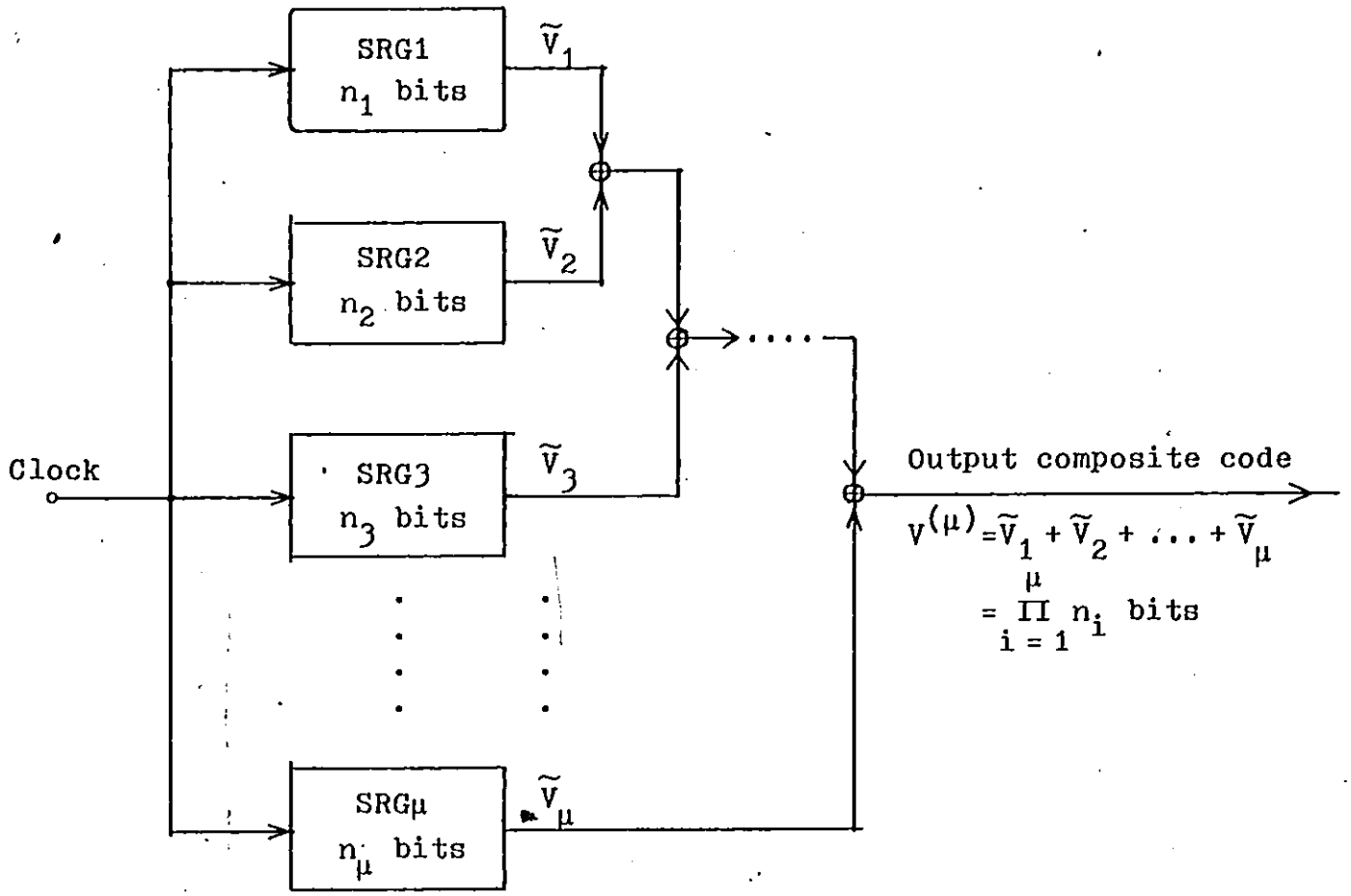


Figure 3.2.2: Typical generalized composite code generator configuration.

COROLLARY 3.2.2:

If V_i is an (n_i,k_i) cyclic code with the generator polynomial $g_i(X)$, $i=1, 2, \ldots, \mu$, and $(n_i,n_j)=1$ for $i\neq j$, then $V^{(\mu)}$ made up of V_i's is cyclic with the generator polynomial

$$g^{(\mu)}(X) = \text{GCD}\left\{\frac{g_1(X)(1+X^n)}{1+X^{n_1}}, \frac{g_2(X)(1+X^n)}{1+X^{n_2}}, \ldots, \frac{g_\mu(X)(1+X^n)}{1+X^{n_\mu}}\right\}, \qquad (3.2.8)$$

where $n=n_1n_2\ldots n_\mu$.

Concerning the weight distribution of composite code $V^{(2)}$ generated by the generator polynomial $g^{(2)}(X)$ of (3.2.7), let us consider the following. Suppose $V_1(X)$ have its j-th coordinate be v_{1j}, $j=0, 1, 2, \ldots, n_1-1$, i.e.,

$$V_1(X) = v_{10}+v_{11}X+\ldots+v_{1,n_1-1}X^{n_1-1}, \qquad (3.2.9)$$

or in vector form,

$$v_1 = (v_{10}, v_{11}, \ldots, v_{1,n_1-1}). \qquad (3.2.10)$$

Similarly, suppose $V_2(X)$ have its code vector of the form

$$v_2 = (v_{20}, v_{21}, \ldots, v_{2,n_2-1}). \qquad (3.2.11)$$

Let $\tilde{v}_{1j}$ be the n_2-tuple formed by repeating v_{1j} n_2 times,

$$\tilde{v}_{1j} = (v_{1j}, v_{1j}, \cdots, v_{1j}), \tag{3.2.12}$$

and let $\hat{v}_j$ be the n_2-tuple defined by

$$\begin{aligned}\hat{v}_j &= \tilde{v}_{1j} \oplus v_2 \\ &= (v_{1j}\oplus v_{20}, v_{1j}\oplus v_{21}, \cdots, v_{1j}\oplus v_{2,n_2-1}).\end{aligned} \tag{3.2.13}$$

Then, on the basis of the nature of ϕ_1 and ϕ_2, and since n_1 and n_2 are relatively prime, we can conclude that the composite codeword $\phi_1(v_1)\oplus\phi_2(v_2)$ is some permutation of the n-tuple $\hat{v}_0\hat{v}_1\hat{v}_2\cdots\hat{v}_{n_1-1}$ and, therefore both of them have the same weight. Now, from (3.2.13), we know that

1). if $v_{1j}=0$, then $\hat{v}_j=v_2$ and have the weight of w_2, the number of "1" in v_2;

2). if $v_{1j}=1$, then $\hat{v}_j=\bar{v}_2$, the complement of v_2, and have the weight of (n_2-w_2), the number of "0" in v_2;

3). within $n=n_1n_2$ bits, the possible number of $v_{1j}=0$ is (n_1-w_1) and the possible number of $v_{1j}=1$ is w_1.

Combining the above conditions, we conclude that the n-tuple $\hat{v}_0\hat{v}_1\hat{v}_2\cdots\hat{v}_{n_1-1}$ has the weight of $w_1(n_2-w_2)+w_2(n_1-w_1)$, and this is the same weight of the composite codeword $\phi_1(v_1)\oplus\phi_2(v_2)$. Hence we have the following theorem.

THEOREM 3.2.3:

With V_1, V_2 and $V^{(2)}$ as previously defined, if $v_i \in V_i$ has weight w_i, i=1, 2, then the corresponding word $\phi_1(v_1)\oplus\phi_2(v_2)$ of $V^{(2)}$ has weight

$$W^{(2)} = w_1(n_2-w_2) + w_2(n_1-w_1). \tag{3.2.14}$$

As a direct application of Theorem 3.2.3, consider the case of M-sequence component codes. As we had mentioned in Section 2.3, a binary M-sequence code is an $(n=2^m-1,k=m)$ cyclic code, all of its nonzero $n=2^m-1$ words having the same weight of $(n+1)/2=2^{m-1}$. Therefore, if V_1 and V_2 are respectively (n_1,k_1) and (n_2,k_2) M-sequence codes with $(n_1,n_2)=1$, then the weight distribution of V_1 is given by

$$\begin{aligned} &N_1(0) = 1, \\ &N_1\left(\frac{n_1+1}{2}\right) = n_1, \end{aligned} \tag{3.2.15a}$$

and that of V_2 is given by

$$\begin{aligned} &N_2(0) = 1, \\ &N_2\left(\frac{n_2+1}{2}\right) = n_2. \end{aligned} \tag{3.2.15b}$$

With the above possible weight combinations, a direct

application of Theorem 3.2.3 will give the weight of $V^{(2)}$ be

$$W^{(2)} = \begin{cases} 0 & , \text{ if } w_1=0,\ w_2=0; \\ n_1(n_2+1)/2, & \text{if } w_1=0,\ w_2=(n_2+1)/2; \\ n_2(n_1+1)/2, & \text{if } w_1=(n_1+1)/2,\ w_2=0; \\ (n_1n_2-1)/2, & \text{if } w_1=(n_1+1)/2,\ w_2=(n_2+1)/2. \end{cases} \tag{3.2.16}$$

Since with respect to Theorem 3.2.1, $\alpha=0$ or $k=k_1+k_2$ for case of V_1 and V_2 are both M-sequence codes, therefore, from (3.2.15a) --- (3.2.16) and follow the proof procedures of Theorem 3.2.3, we obtain the following weight distribution of code $V^{(2)}$.

THEOREM 3.2.4:

If V_i is an (n_i,k_i) M-sequence code for i=1, 2, n_1 and n_2 being relatively prime, then the composite code $V^{(2)}$ has the weight distribution

$$\begin{aligned} N(0) &= 1, \\ N\left(\frac{n_1n_2-1}{2}\right) &= n_1n_2, \\ N\left(\frac{n_1n_2+n_1}{2}\right) &= n_1, \\ N\left(\frac{n_1n_2+n_2}{2}\right) &= n_2, \end{aligned} \tag{3.2.17}$$

where N(w) is the number of words of weight w.

Generally, in the case of μ component codes, the weight equation of code $V^{(\mu)}$ is

$$W^{(\mu)} = W^{(\mu-1)}(n_\mu - w_\mu) + w_\mu(N^{(\mu-1)} - W^{(\mu-1)}), \tag{3.2.18}$$

where $W^{(\mu-1)}$ and $N^{(\mu-1)} = n_1 n_2 \cdots n_{\mu-1}$ are respectively the weight and the length of the composite code $V^{(\mu-1)}$, while w_μ and n_μ are respectively those of the μ-th component code V_μ.

The importance of Theorem 3.2.3 or Equ.(3.2.18) lies in that, given weight distributions of V_i's, that of $V^{(\mu)}$ can be computed easily. This also means, of course, that the weight distribution of the dual of $V^{(\mu)}$ can be determined using the famous MacWilliams identities [18, Chapter 5].

Next we consider the question of minimum distance of the code $V^{(2)}$ of (3.2.2) in general. Remember that the minimum distance of a linear code is equal to the minimum weight of the nonzero code vectors.

THEOREM 3.2.5:

Let d_i and D_i be the minimum and maximum distances respectively in V_i for i=1, 2. Then

$$\begin{aligned} d &= \min[n_2 d_1,\ n_1 d_2,\ n_2 D_1 + n_1 D_2 - 2D_1 D_2], \\ D &= \max[n_2 D_1,\ n_1 D_2,\ n_2 D_1 + n_1 D_2 - 2D_1 D_2], \end{aligned} \tag{3.2.19}$$

where d and D are respectively the minimum and maximum distances in the composite code $V^{(2)}$.

The proof of the above theorem is given in [27]; it is based on the extremity property of the normalized weight line segment.

A recursive application of Theorem 3.2.5 for the case when all of the μ component codes are M-sequence codes, gives the following:

COROLLARY 3.2.5:

If V_i is an (n_i, k_i) M-sequence code, i=1, 2, ..., μ, with $n_1 < n_2 < \ldots < n_\mu$ and n_i's relatively prime pairwise, then the composite code $V^{(\mu)}$ has

$$\begin{aligned} d &= \left(\frac{n_1 n_2 - 1}{2}\right) n_3 n_4 \ldots n_\mu, \\ D &= \left(\frac{n_1 + 1}{2}\right) n_2 n_3 \ldots n_\mu, \end{aligned} \tag{3.2.20}$$

where d and D are respectively the minimum and maximum distances in $V^{(\mu)}$.

3.2.2 CORRELATION SPECTRA OF GENERALIZED COMPOSITE CODES

Now, let us go to examine the correlation between words of generalized composite code $V^{(\mu)}$ for the case when all of the component codes V_i are M-sequence codes. To start with, we recall that an $(n_i=2^{m_i}-1, k_i=m_i)$ binary M-sequence code V_i has n_i nonzero codewords all of which have the same weight of $2^{m_i-1}=(n_i+1)/2$. We also recall that the correlation between two binary n-tuples u and v is given by

$$\theta(u,v) = \frac{n-2\cdot dis(u,v)}{n} = \frac{n-2\cdot|u \oplus v|}{n}, \qquad (3.2.21)$$

where dis(u,v) is the distance between u and v, and $|u\oplus v|$ is the weight of codeword $u\oplus v$. For a linear code V, if both u and v are codewords of V, then $u\oplus v$ is also a word of V.

Let S_μ be the set of correlations in $V^{(\mu)}$. For example, as we have known, $S_1=\{1,-1/n_1\}$ for M-sequence code. With reference to (3.2.18), we note that after j M-sequence component codes have been combined into $V^{(j)}$, $j \in \{1, 2, \ldots, \mu-1\}$, if we add another (j+1)th M-sequence code which has a length relatively prime to those of previous j component codes, then we will get

$$W^{(j+1)} = W^{(j)}(n_{j+1}-w_{j+1}) + w_{j+1}(N^{(j)}-W^{(j)}), \qquad (3.2.22)$$

where $W^{(j)}$ is a weight in code $V^{(j)}$, $W^{(j+1)}$ is a weight in code $V^{(j+1)}$, and $N^{(j)}=n_1n_2\cdots n_j$ is the length of $V^{(j)}$. Surely, the length of code $V^{(j+1)}$ is

$$N^{(j+1)} = N^{(j)}n_{j+1} = n_1n_2\cdots n_{j+1}. \tag{3.2.23}$$

Since the possible weight of component code V_{j+1} is $w_{j+1}=0$ or $w_{j+1}=(n_{j+1}+1)/2$, therefore (3.2.22) gives

$$W^{(j+1)} = \begin{cases} W^{(j)}n_{j+1}, & \text{if } w_{j+1}=0; \quad (3.2.24a) \\ \dfrac{N^{(j+1)}+N^{(j)}-2W^{(j)}}{2}, & \text{if } w_{j+1}=\dfrac{n_{j+1}+1}{2}. \quad (3.2.24b) \end{cases}$$

Corresponding to (3.2.24a), we have

$$\begin{aligned} \theta^{(j+1)} &= \frac{N^{(j+1)} - 2W^{(j+1)}}{N^{(j+1)}} \\ &= \frac{N^{(j+1)} - 2W^{(j)}n_{j+1}}{N^{(j+1)}} \\ &= \frac{N^{(j)} - 2W^{(j)}}{N^{(j)}}, \end{aligned}$$

so that

$$\theta^{(j+1)} = \theta^{(j)}. \tag{3.2.25a}$$

Corresponding to (3.2.24b), we have

$$\theta^{(j+1)} = \frac{N^{(j+1)} - 2W^{(j+1)}}{N^{(j+1)}}$$

$$= \frac{N^{(j+1)} - N^{(j+1)} - N^{(j)} + 2W^{(j)}}{N^{(j+1)}}$$

$$= \frac{-1}{N^{(j+1)}} \left(N^{(j)} - 2W^{(j)}\right)$$

$$= -\frac{1}{n_{j+1}} \left(\frac{N^{(j)} - 2W^{(j)}}{N^{(j)}}\right),$$

so that

$$\theta^{(j+1)} = -\frac{1}{n_{j+1}} \theta^{(j)}. \qquad (3.2.25b)$$

What (3.2.25a) and (3.2.25b) imply is that if a certain correlation value θ belongs to S_j, then S_{j+1} contains both θ and $-\theta/n_{j+1}$.

Recalling that

$$S_1 = \{1, -\frac{1}{n_1}\}, \qquad (3.2.26a)$$

and using (3.2.25a) and (3.2.25b) recursively, we see that

$$S_2 = S_1 \cup \frac{-1}{n_2} S_1$$

$$= \{1, -\frac{1}{n_1}\} \cup \frac{-1}{n_2}\{1, -\frac{1}{n_1}\}$$

$$= \{1, -\frac{1}{n_1}, -\frac{1}{n_2}, \frac{1}{n_1 n_2}\}, \qquad (3.2.26b)$$

$$S_3 = S_2 \cup \frac{-1}{n_3} S_2$$

$$= \{1, \frac{-1}{n_1}, \frac{-1}{n_2}, \frac{1}{n_1 n_2}\} \cup \frac{-1}{n_3}\{1, \frac{-1}{n_1}, \frac{-1}{n_2}, \frac{1}{n_1 n_2}\}$$

$$= \{1, \frac{-1}{n_1}, \frac{-1}{n_2}, \frac{-1}{n_3}, \frac{1}{n_1 n_2}, \frac{1}{n_1 n_3}, \frac{1}{n_2 n_3}, \frac{-1}{n_1 n_2 n_3}\},$$

(3.2.26c)

and so on. Examining S_1, S_2, S_3, ..., we conclude the following:

THEOREM 3.2.6:

If the component codes V_i are all $(n_i = 2^{m_i} - 1, k_i = m_i)$ binary M-sequence codes, where $i = 1, 2, \ldots, \mu$ and $(n_i, n_j) = 1$, then the generalized composite code $V_r^{(\mu)}$ has the set S_μ, of correlations, defined by $S_\mu = \sigma_0 \cup \sigma_1 \cup \sigma_2 \cup \ldots \cup \sigma_\mu$, where σ_0 is the set containing the only number of 1, and σ_j, for $j \geq 1$, is the set of numbers of the form $(-1)^j / n_{i_1} n_{i_2} \ldots n_{i_j}$.

To clarify the statement of the Theorem 3.2.6, we mention, as an example, that, if $\mu=5$, then, say,

$$\sigma_4 = \{ \frac{(-1)^4}{n_1n_2n_3n_4}, \frac{(-1)^4}{n_1n_2n_3n_5}, \frac{(-1)^4}{n_1n_2n_4n_5}, \frac{(-1)^4}{n_1n_3n_4n_5}, \frac{(-1)^4}{n_2n_3n_4n_5} \}.$$

Figures 3.2.3 and 3.2.4 give examples of the possible correlation values and their corresponding power spectra for the generalized composite codes $V^{(2)}$ and $V^{(3)}$, respectively. In Figure 3.2.3, we use $(n_1=3,k_1=2)$ and $(n_2=7,k_2=3)$ M-sequence codes as $V^{(2)}$'s component codes, while in Figure 3.2.4, $(n_1=3,k_1=2)$, $(n_2=7,k_2=3)$, and $(n_3=31,k_3=5)$ M-sequence component codes are used. In either case, note that $(n_i,n_j)=1$ for $i \neq j$. By cyclic shifting a typical codeword $V^{(2)}(X) \in V^{(2)}$ ($V^{(3)}(X) \in V^{(3)}$) and counting the Hamming distance between $X^lV^{(2)}(X)$ ($X^lV^{(3)}(X)$) and $V^{(2)}(X)$ ($V^{(3)}(X)$) for $l = 0, 1, 2, \ldots, n-1$, where $n=n_1n_2$ ($n=n_1n_2n_3$) is the code length of $V^{(2)}$ ($V^{(3)}$), then through the operation of (3.2.21) we obtain the typical autocorrelation function of Figure 3.2.3(a) (Figure 3.2.4(a)), which obviously coincides with what we had derived in Theorem 3.2.6.

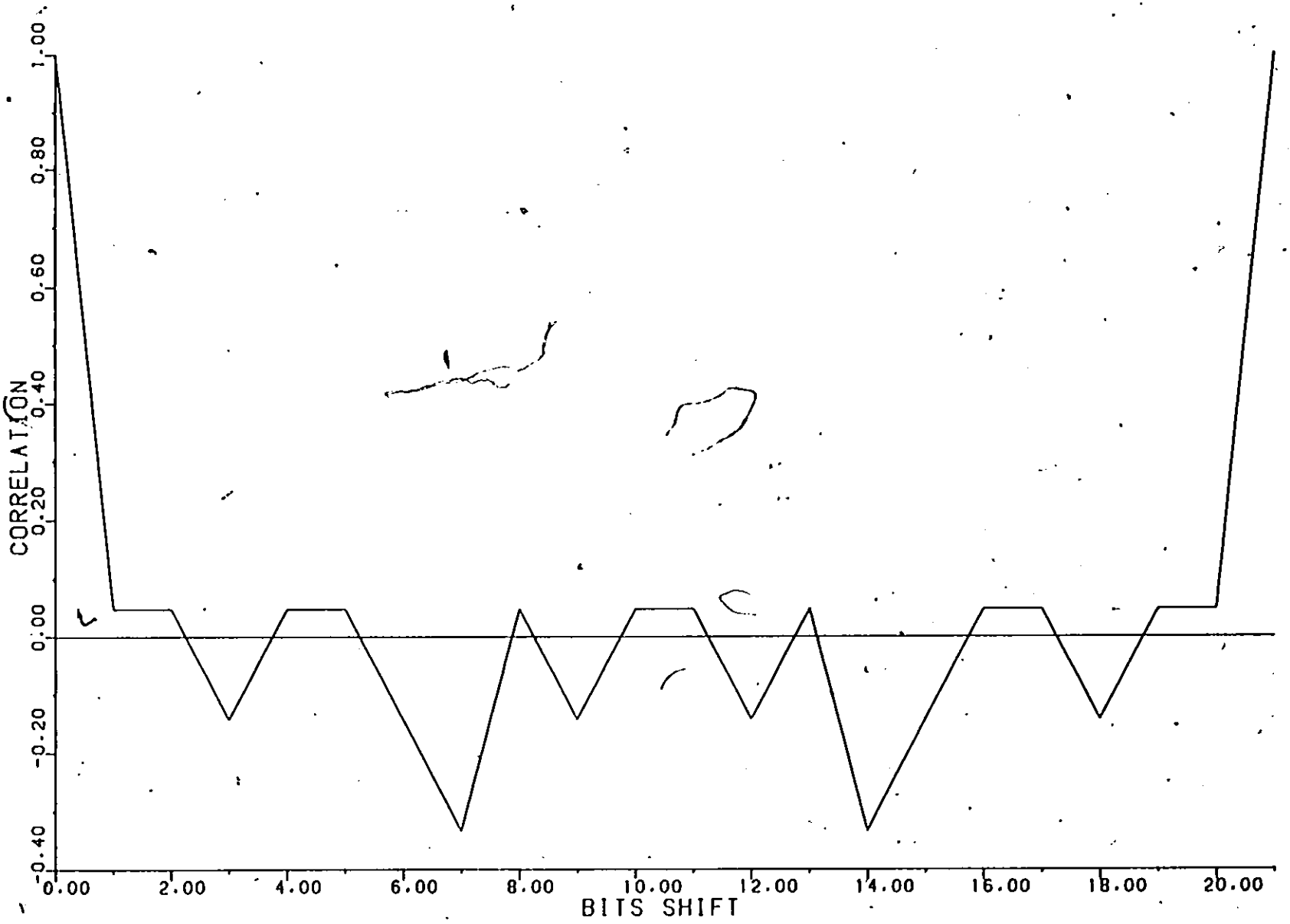


Figure 3.2.3(a): Autocorrelation function of the typical generalized composite code $V^{(2)}$.

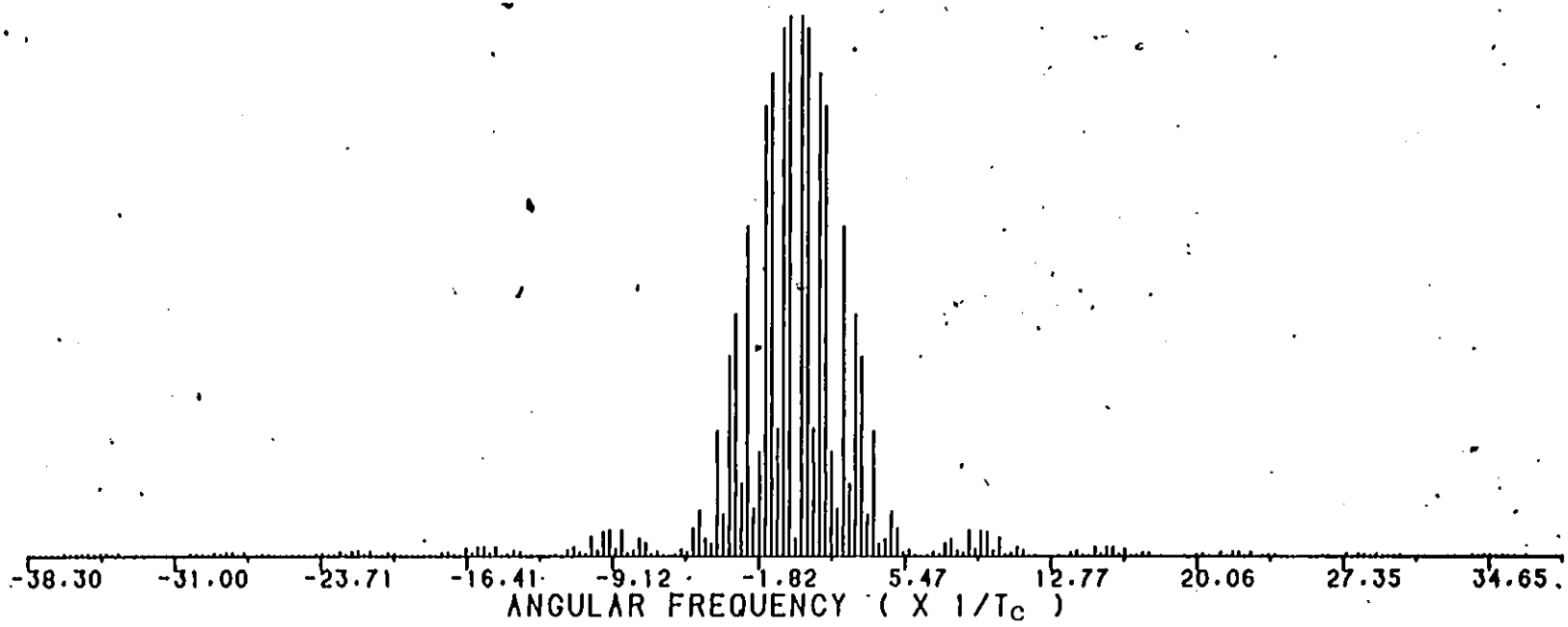


Figure 3.2.3(b): Power spectrum corresponding to the function of Figure 3.2.3(a).

Corresponding to the autocorrelation functions of Figures 3.2.3(a) and 3.2.4(a); the power spectra of Figures 3.2.3(b) and 3.2.4(b) are obtained from the well-known Fourier Transform relationship between correlation function and power spectrum. Alternatively, since the discussed autocorrelation function can be represented as the sum of several partial functions all of the same form but with different periods and amplitudes, the power spectrum can be approximated as the sum of several partial spectra [10], each corresponds to a certain partial correlation function. In any event, the resulted power spectrum is line spectrum which bears much resemblances in envelopes with that of M-sequence code (refer to Figure 2.3.1(b)). However, as we can see, the amplitudes are actually "lumpy". This is because the based correlation functions having minor peaks within out-of-phase durations.

From Theorem 3.2.6 together with Figures 3.2.3(a) and 3.2.4(a) it is clear that, among the correlations $\theta \neq 1$, $|\theta| \leq 1/n_1$, if we assume, without losing any generality, that $n_1 < n_2 < \ldots < n_\mu$. Thus, when the component codes are all M-sequence codes, the magnitude of the correlation in the generalized composite code $V^{(\mu)}$ can be kept below as small a value as desired by making the length of the shortest component code long enough.

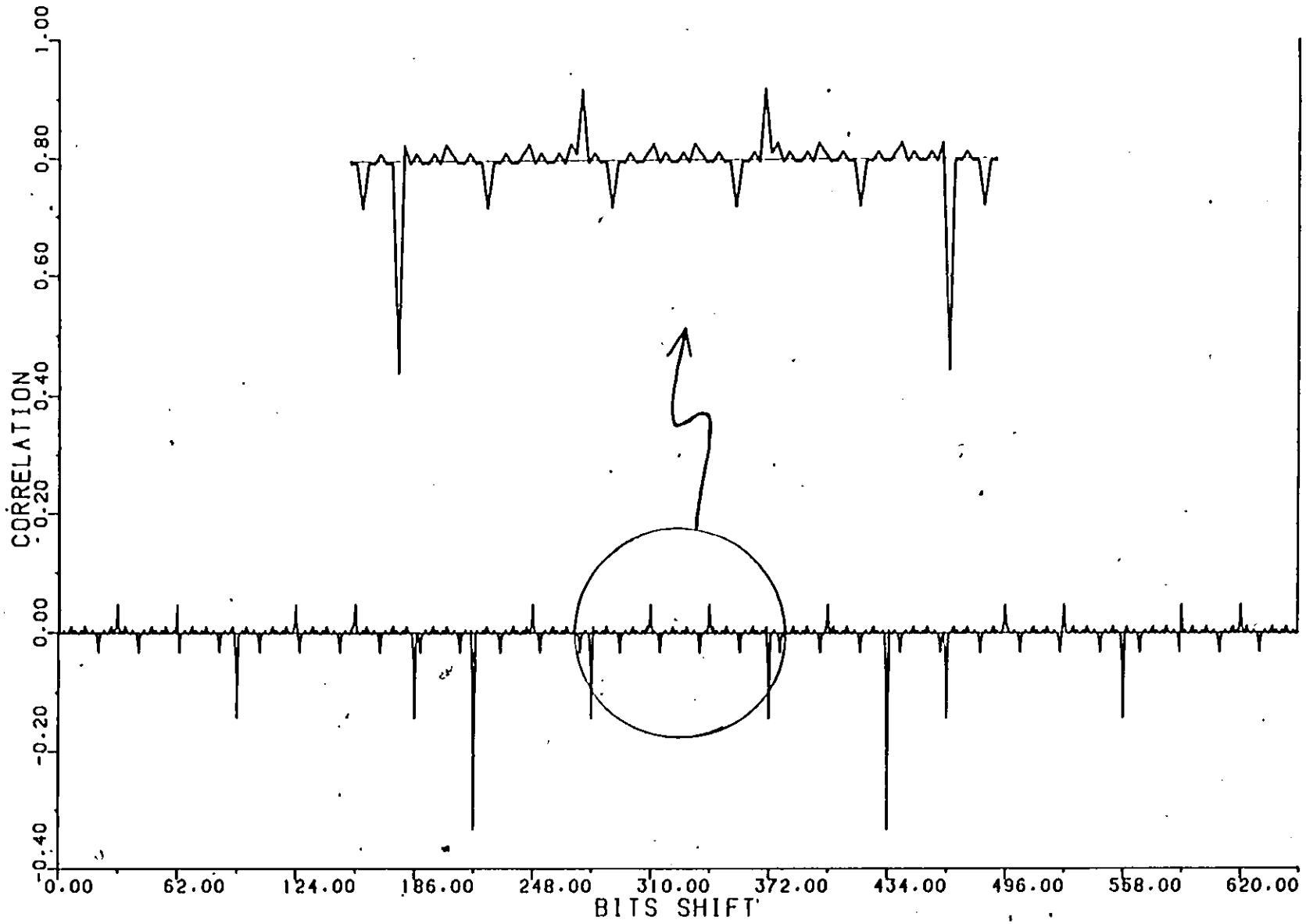


Figure 3.2.4(a): Autocorrelation function of the typical generalized composite code $V^{(3)}$.

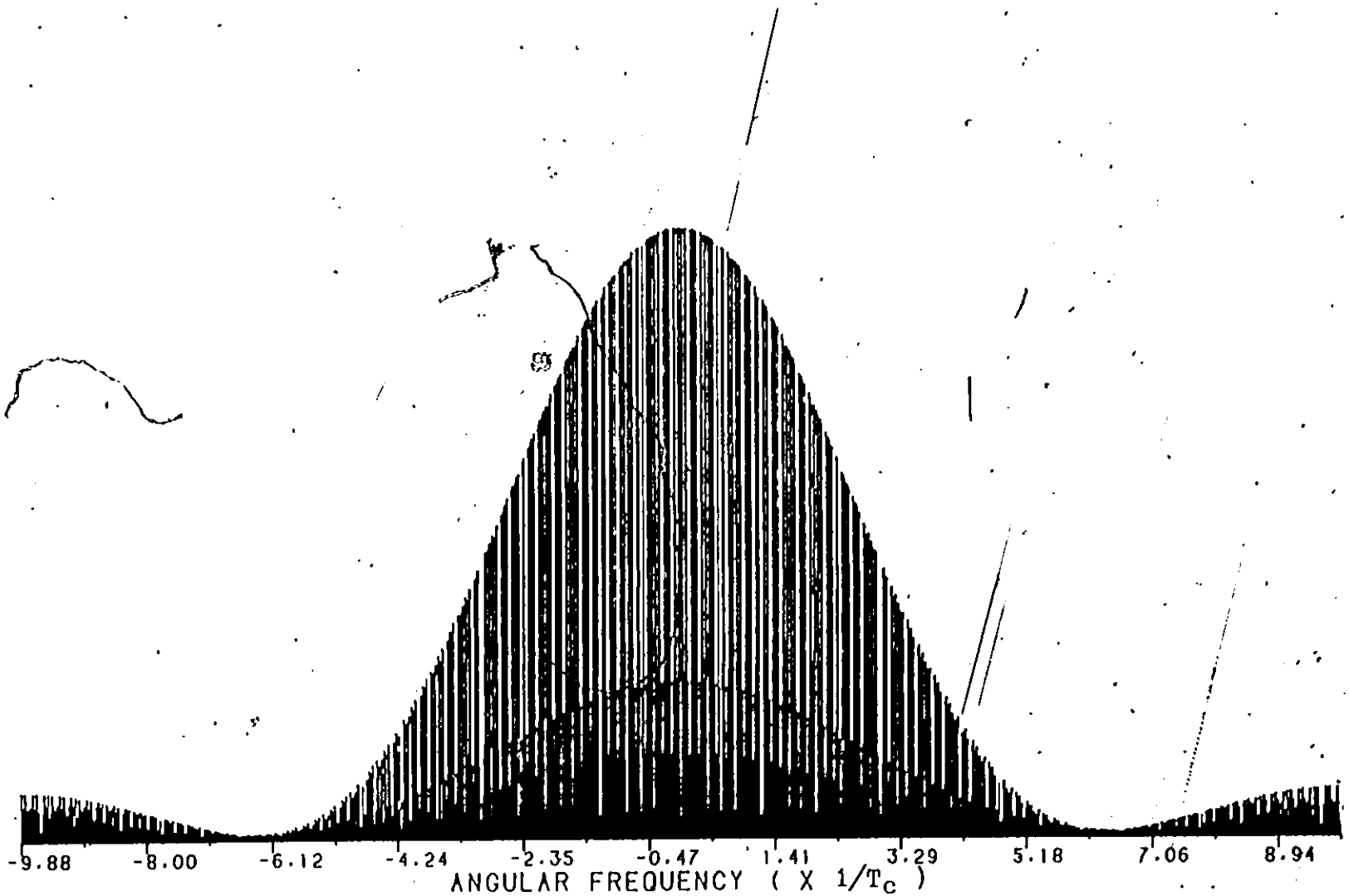


Figure 3.2.4(b): Power spectrum corresponding to the function of Figure 3.2.4(a).

Having derived the possible correlation values of generalized composite codes, we now proceed to discuss the question as to the number of positions by which a codeword has to be shifted so that the correlation between the codeword and its cyclic shift is any specific value from the set of values given in Theorem 3.2.6.

Let V_A and V_B be two binary cyclic codes, V_A being (n_A,k_A) and V_B being (n_B,k_B), where $(n_A,n_B) = 1$. Let V_C be the composite code, as defined in (3.2.2), with V_A and V_B as the component codes, so that a codeword $V_C(X)$ of V_C is given by

$$V_C(X) = V_A(X)\frac{1+X^{n_An_B}}{1+X^{n_A}} + V_B(X)\frac{1+X^{n_An_B}}{1+X^{n_B}}, \quad (3.2.27)$$

where $V_A(X)$ belongs to V_A and $V_B(X)$ belongs to V_B. Now let us consider

$$F(X) = V_C(X)+X^{\lambda}V_C(X) \bmod (1+X^{n_An_B}). \quad (3.2.28)$$

If $\lambda = Mn_A$, then from (3.2.27) and (3.2.28), we have

$$F(X) = V_B(X)\frac{1+X^{n_An_B}}{1+X^{n_B}}(1+X^{Mn_A}) \bmod (1+X^{n_An_B}), \quad (3.2.29)$$

which can be rewritten as

$$F(X) = \beta(X)+X^{Mn_A}\beta(X) \bmod (1+X^{n_An_B}), \qquad (3.2.30)$$

where $\beta(X)$ is the (n_An_B)-tuple obtained by repeating $V_B(X)$ n_A times, i.e., $\beta(X) = V_B(X)\{(1+X^{n_An_B})/(1+X^{n_B})\}$. But shifting $\beta(X)$ cyclically by Mn_A positions is the same as shifting $V_B(X)$ cyclically by λ_B positions, where $\lambda_B = Mn_A$ modulo n_B.

This means from (3.2.27) to (3.2.30), that, if $\lambda = Mn_A$, $0 < M < n_B$, then

$$\begin{aligned} &\left| V_C(X)+X^{Mn_A}V_C(X) \bmod (1+X^{n_An_B})\right| \\ &= \left| V_B(X)+X^{\lambda_B}V_B(X) \bmod (1+X^{n_B})\right|\cdot n_A. \end{aligned} \qquad (3.2.31)$$

Setting

$$\left| V_B(X)+X^{\lambda_B}V_B(X) \bmod (1+X^{n_B})\right| = \Delta, \qquad (3.2.32)$$

we see that

$$\begin{aligned} &\theta(V_B(X), X^{\lambda_B}V_B(X) \bmod (1+X^{n_B})) \\ &= \frac{n_B - 2\Delta}{n_B} \end{aligned} \qquad (3.2.33)$$

and

$$\theta(V_C(X),\ X^{Mn_A}V_C(X) \bmod (1+X^{n_An_B})) = \frac{n_An_B - 2n_A\Delta}{n_An_B} . \qquad (3.2.34)$$

Comparing (3.2.33) and (3.2.34) we see that they are equal. From this we conclude the following:

THEOREM 3.2.7:

Suppose V_A and V_B are two binary cyclic codes with the respective lengths n_A and n_B such that $(n_A, n_B) = 1$. Then every correlation occurring in V_A (V_B), for a cyclic shift of λ_A (λ_B) positions, occurs also in the composite code V_C for a cyclic shift of Mn_B (Mn_A) positions, where M and λ_A (λ_B) are related through $\lambda_A = Mn_B$ modulo n_A ($\lambda_B = Mn_A$ modulo n_B).

A composite code $V^{(\mu)}$ with component codes $V_1, V_2, \ldots, V_\mu$ can be treated as a composite code V_C with component codes V_A and V_B, where V_A is a composite code with component codes, say, $V_{i_1}, V_{i_2}, \ldots, V_{i_u}$ and V_B is a composite code with component codes $V_{i_{u+1}}, V_{i_{u+2}}, \ldots, V_{i_{u+(\mu-u)}}$, where $V_j \in \{V_1, V_2, \ldots, V_\mu\}$. Using this fact with reference to Theorems 3.2.6 and 3.2.7 in a recursive fashion, we obtain the following:

THEOREM 3.2.8:

With reference to Theorem 3.2.6, the correlation $(-1)^a/n_{i_1}n_{i_2}\ldots n_{i_a}$ occurs for a cyclic shift of $bn/n_{i_1}n_{i_2}\ldots n_{i_a}$ positions, where b is a positive integer and not a multiple of $n_{i_1}, n_{i_2}, \ldots, n_{i_a}$, a is a positive integer ≥ 1 and $\leq \mu-1$, and $n = n_1n_2\ldots n_\mu$.

With respect to Theorem 3.2.8 we note that for other cyclic shifts we have the correlation of $(-1)^\mu/n$.

To clarify the statement of Theorem 3.2.8, suppose $\mu=4$. Then according to Theorem 3.2.6, $V^{(4)}$ has the set S_4 of correlations given by

$$S_4 = \{1, \frac{-1}{n_1}, \frac{-1}{n_2}, \frac{-1}{n_3}, \frac{-1}{n_4}, \frac{1}{n_1n_2}, \frac{-1}{n_1n_3}, \frac{1}{n_1n_4}, \frac{1}{n_2n_3}, \frac{1}{n_2n_4},$$

$$\frac{1}{n_3n_4}, \frac{-1}{n_1n_2n_3}, \frac{-1}{n_1n_2n_4}, \frac{-1}{n_1n_3n_4}, \frac{-1}{n_2n_3n_4}, \frac{1}{n_1n_2n_3n_4}\}.$$

If λ is the number of positions by which the codeword is cyclically shifted, then, according to Theorem 3.2.8, the correlation of 1 occurs for $\lambda = 0$, $-1/n_i$ for $\lambda = bn_1n_2n_3n_4/n_i$, $1/n_in_j$ for $\lambda = bn_1n_2n_3n_4/n_in_j$, $-1/n_in_jn_k$ for $\lambda = bn_1n_2n_3n_4/n_in_jn_k$, and $1/n_1n_2n_3n_4$ elsewhere. We note here that b is not a multiple of any of the numbers in the denominator for a given λ.

Chapter IV

CONCLUDING REMARKS

Two families of composite codes with good correlation properties have been examined in this thesis. These composite codes are combined in bit-by-bit modulo-2 addition fashions from some suitable M-sequence component codes. Both of them are nonmaximal-length cyclic codes and both contain larger number of code sequences. It is these bounded correlation magnitudes and larger codes numbers available which make the proposed composite codes attractive in the recent spread spectrum and multiple access communications applications.

Several aspects for developing our correlation studies of the above composite codes were inclusively discussed in the first several sections. We introduced the fundamental concept of finite field theory, the code generation configuration, the general properties of cyclic codes, and especially the basic characteristics of binary M-sequence codes. With these foundations, the structures and the correlation properties of our composite codes were shown to have intimate relationship with those of their component codes.

For the first class of composite codes, it was showed that the Gold codes of length $n=2^m-1$ are actually code collections of two preferred M-sequence codes and n nonmaximal-length cyclic codes, with the latter be summed from the different relative phase-shifts of the former. By applying the idealized 2-level auto-correlation function and the preferred 3-level or 4-level cross-correlation function inherented in the preferred pairs of M-sequence base codes, Gold codes were constructed to meet those well-bounded correlation spectra. More precisely, the peak correlation magnitude of a set of Gold codes of length $n=2^m-1$ is bounded by

$$|\theta(m)| \leq \begin{cases} (2^{\lfloor (m+2)/2 \rfloor}+1)/n, & \text{for } m \not\equiv 0 \bmod 4; \\ (2^{(m+2)/2}-1)/n, & \text{for } m \equiv 0 \bmod 4. \end{cases}$$

For large enough n value, it is seen that the correlation bounds of Gold codes tend to be $\sqrt{2}/\sqrt{n}$ or $2/\sqrt{n}$.

For the generalized composite code, several M-sequence codes with pairwise relatively prime lengths are modulo-2 combined. If n_i and k_i, $i=1, 2, \ldots, \mu$, are respectively the length and the dimension of the M-sequence codes V_i, where $(n_i,n_j)=1$ for $i\neq j$, then the generalized composite code $V^{(\mu)}$ made up of these μ M-sequence component codes has a length of $n=\prod_{i=1}^{\mu} n_i$ and a dimension of $k=\sum_{i=1}^{\mu} k_i$. From this dimensionality, it was deduced that the possible codewords

number of the above generalized composite code is $2^k = \prod_{i=1}^{\mu}(1+n_i)$. These words are distributed among $2^\mu = \sum_{i=0}^{\mu}\binom{\mu}{i}$ cyclic classes. The minimum distance, maximum distance, and weight distribution of generalized composite code were also well-defined from those of its component codes. Based on these parameters, it was shown that, if n_1 is the shortest M-sequence component code length, the peak correlation magnitude of generalized composite code is well-bounded by $1/n_1$. Moreover, the number of positions by which a codeword has to be shifted so that the correlation between the codeword and its cyclic shift is any specific value in the set of correlation spectrum was also well developed. This result is very important in expediting the correlation detection of coded messages.

From the above mentioned correlation bounds, it is seen that the code lengths of the discussed composite codes have to be relatively large to have the bounded correlation magnitudes be as small as possible. This incurs the practical constraints of maximum available number of memory cells for storing the referenced correlating code sequences and hence the speed of code acquisitions. Partial correlation is thus an attractive consideration under such constraints. However, unlike the periodic correlation functions, the partial correlation of two periodic code sequences is more random and more intractable. Although

some researchers have formulated certain statistical results [4], [6], [19], [30], they still are not satisfying ones. Thus, partial correlations or fast acquisition algorithms of the discussed composite codes are worth to take a further study.

As a final remark, we point out that the composite codes we have considered in this thesis are both made up of M-sequence component codes. Although they possess well-bounded correlation magnitudes, the available numbers of codes are still not large enough for the modern multiple access applications. From the construction fashions of these composite codes, we wonder whether there are some other codes which can be taken as our component codes and result in an even larger and even bounded code set. For example, it seems that Gold codes can be taken into our generalized composite codes to result in a larger code set which has the same correlation bounds of Gold codes selves. The detail of this possibility remains to be investigated.

Appendix A

PRIMITIVE POLYNOMIAL PAIRS FOR CONSTRUCTING GOLD CODES

The following primitive binary polynomials of degree m, m=3 to 13, can be used to generate M-sequence codes. Any preferred pair of these polynomials can be used to construct a set of Gold composite codes. For convenience, we call each polynomial $P_i(X)$ if $\alpha^i \in GF(2^m)$ is the least exponent root of $P_i(X)$. Hence every primitive polynomial of the left column is named by the first entry of the preferred pair of polynomials in the right column. For example, in the case of m=6, we have $P_1(X)=1+X+X^6$, $P_5(X)=1+X+X^2+X^5+X^6$, $P_{11}(X)=1+X^2+X^3+X^5+X^6$, and so on.

Corresponding to each primitive polynomial $P_i(X)$, we can base the logic $P_i(\beta)=0$, where $\beta=\alpha^i$, to construct a version of $GF(2^m)$. Among elements β^k's, k=1, 2, ..., 2^m-2, the preferred polynomial pair $[P_i(X),P_j(X)]$ occurred when $\beta(=\alpha^i)$ and $\beta^\lambda(=\alpha^{\lambda i} \bmod \alpha^{2^m-1}=\alpha^j)$ are respectively roots of $P_i(X)$ and $P_j(X)$, where $\lambda = 2^{\lfloor (m+2)/2 \rfloor}+1$ for $m \not\equiv 0 \bmod 4$ and $\lambda = 2^{(m+2)/2}-1$ for $m = 0 \bmod 4$, as established in Sections 3.1.2 and 3.1.3.

DEGREE 3

1 +X1 +X3	(P 1 , P 3)
1 +X2 +X3	(P 3 , P 1)

DEGREE 4

1 +X1 +X4	(P 1 , P 7)
1 +X3 +X4	(P 7 , P 1)

DEGREE 5

1 +X2 +X5	(P 1 , P 5)
1 +X2 +X3 +X4 +X5	(P 3 , P15)
1 +X1 +X2 +X4 +X5	(P 5 , P 7)
1 +X1 +X2 +X3 +X5	(P 7 , P 1)
1 +X1 +X3 +X4 +X5	(P11 , P 3)
1 +X3 +X5	(P15 , P11)

DEGREE 6

1 +X1 +X6	(P 1 , P 5)
1 +X1 +X2 +X5 +X6	(P 5 , P11)
1 +X2 +X3 +X5 +X6	(P11 , P31)
1 +X1 +X3 +X4 +X6	(P13 , P 1)
1 +X1 +X4 +X5 +X6	(P23 , P13)
1 +X5 +X6	(P31 , P23)

DEGREE 7

1 +X3 +X7	(P 1 , P 9)
1 +X1 +X2 +X3 +X7	(P 3 , P 27)
1 +X2 +X3 +X4 +X7	(P 5 , P 43)
1 +X1 +X2 +X4 +X5 +X6 +X7	(P 7 , P 63)
1 +X1 +X2 +X3 +X4 +X5 +X7	(P 9 , P 13)
1 +X2 +X4 +X6 +X7	(P 11 , P 15)
1 +X1 +X7	(P 13 , P 47)
1 +X1 +X2 +X3 +X5 +X6 +X7	(P 15 , P 1)
1 +X1 +X3 +X6 +X7	(P 19 , P 11)
1 +X2 +X5 +X6 +X7	(P 21 , P 31)
1 +X6 +X7	(P 23 , P 5)
1 +X1 +X4 +X6 +X7	(P 27 , P 29)
1 +X1 +X3 +X5 +X7	(P 29 , P 7)
1 +X4 +X5 +X6 +X7	(P 31 , P 19)
1 +X1 +X2 +X5 +X7	(P 43 , P 3)
1 +X3 +X4 +X5 +X7	(P 47 , P 21)
1 +X2 +X3 +X4 +X5 +X6 +X7	(P 55 , P 23)
1 +X4 +X7	(P 63 , P 55)

DEGREE 8

1 +X2 +X3 +X4 +X8	(P 1 , P 31)
1 +X3 +X5 +X6 +X8	(P 7 , P 59)
1 +X1 +X2 +X5 +X6 +X7 +X8	(P 11 , P 43)
1 +X1 +X3 +X5 +X8	(P 13 , P 37)
1 +X2 +X5 +X6 +X8	(P 19 , P 61)
1 +X1 +X5 +X6 +X8	(P 23 , P 47)
1 +X2 +X3 +X7 +X8	(P 29 , P 13)
1 +X2 +X3 +X5 +X8	(P 31 , P 19)
1 +X1 +X2 +X3 +X4 +X6 +X8	(P 37 , P127)
1 +X1 +X6 +X7 +X8	(P 43 , P 29)
1 +X3 +X5 +X7 +X8	(P 47 , P 91)
1 +X1 +X2 +X7 +X8	(P 53 , P 23)
1 +X2 +X3 +X6 +X8	(P 59 , P 11)
1 +X1 +X2 +X3 +X6 +X7 +X8	(P 61 , P 53)
1 +X2 +X4 +X5 +X6 +X7 +X8	(P 91 , P 1)
1 +X4 +X5 +X6 +X8	(P127 , P 7)

DEGREE 9

1 +X4 +X9	(P 1 , P 17)
1 +X3 +X4 +X6 +X9	(P 3 , P 51)
1 +X4 +X5 +X8 +X9	(P 5 , P 85)
1 +X1 +X4 +X8 +X9	(P 9 , P 83)
1 +X2 +X3 +X5 +X9	(P 11 , P187)
1 +X1 +X2 +X4 +X5 +X6 +X9	(P 13 , P183)
1 +X5 +X6 +X8 +X9	(P 15 , P255)
1 +X1 +X3 +X4 +X6 +X7 +X9	(P 17 , P 25)
1 +X2 +X7 +X8 +X9	(P 19 , P 29)
1 +X3 +X5 +X6 +X7 +X8 +X9	(P 23 , P 31)
1 +X1 +X5 +X6 +X7 +X8 +X9	(P 25 , P117)
1 +X1 +X2 +X3 +X7 +X8 +X9	(P 27 , P 95)
1 +X1 +X3 +X5 +X6 +X8 +X9	(P 29 , P223)
1 +X1 +X3 +X4 +X9	(P 31 , P 1)
1 +X1 +X2 +X3 +X5 +X6 +X9	(P 37 , P 59)
1 +X2 +X3 +X6 +X7 +X8 +X9	(P 39 , P 19)
1 +X1 +X4 +X5 +X6 +X8 +X9	(P 41 , P 93)
1 +X1 +X3 +X6 +X7 +X8 +X9	(P 43 , P 55)
1 +X2 +X3 +X4 +X5 +X6 +X9	(P 45 , P127)
1 +X1 +X3 +X4 +X6 +X8 +X9	(P 47 , P 9)
1 +X2 +X4 +X6 +X7 +X8 +X9	(P 51 , P 75)
1 +X2 +X4 +X7 +X9	(P 53 , P 27)
1 +X2 +X3 +X4 +X5 +X7 +X9	(P 55 , P 53)

1 +X2 +X4 +X5 +X6 +X7 +X9	(P 57 , P 87)
1 +X1 +X2 +X3 +X6 +X7 +X9	(P 59 , P123)
1 +X1 +X2 +X3 +X4 +X6 +X9	(P 61 , P 15)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X8 +X9	(P 75 , P191)
1 +X1 +X2 +X6 +X7 +X8 +X9	(P 79 , P 13)
1 +X2 +X4 +X8 +X9	(P 83 , P 23)
1 +X1 +X2 +X4 +X5 +X7 +X9	(P 85 , P125)
1 +X2 +X5 +X7 +X9	(P 87 , P 79)
1 +X3 +X4 +X5 +X6 +X7 +X9	(P 93 , P 3)
1 +X3 +X4 +X5 +X7 +X8 +X9	(P 95 , P 41)
1 +X1 +X2 +X3 +X5 +X7 +X9	(P103 , P109)
1 +X1 +X5 +X7 +X9	(P107 , P 61)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X8 +X9	(P109 , P 5)
1 +X1 +X2 +X3 +X4 +X8 +X9	(P111 , P 43)
1 +X1 +X2 +X3 +X6 +X8 +X9	(P117 , P 57)
1 +X1 +X2 +X7 +X9	(P123 , P 47)
1 +X4 +X6 +X7 +X9	(P125 , P 37)
1 +X3 +X5 +X6 +X9	(P127 , P103)
1 +X2 +X4 +X5 +X7 +X8 +X9	(P171 , P 11)
1 +X1 +X3 +X4 +X5 +X8 +X9	(P183 , P 45)
1 +X3 +X4 +X6 +X7 +X8 +X9	(P187 , P 39)
1 +X1 +X4 +X5 +X9	(P191 , P171)
1 +X1 +X5 +X8 +X9	(P223 , P107)
1 +X2 +X3 +X5 +X6 +X8 +X9	(P239 , P111)
1 +X5 +X9	(P255 , P239)

DEGREE 10

1 +X3 +X10	(P 1 , P 17)
1 +X2 +X3 +X8 +X10	(P 5 , P 85)
1 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X10	(P 7 , P119)
1 +X1 +X2 +X3 +X5 +X6 +X10	(P 13 , P221)
1 +X2 +X3 +X6 +X8 +X9 +X10	(P 17 , P 41)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X8 +X10	(P 19 , P 53)
1 +X1 +X3 +X4 +X10	(P 23 , P 59)
1 +X1 +X5 +X8 +X10	(P 25 , P181)
1 +X4 +X5 +X8 +X10	(P 29 , P379)
1 +X1 +X4 +X9 +X10	(P 35 , P167)
1 +X1 +X5 +X6 +X8 +X9 +X10	(P 37 , P235)
1 +X2 +X5 +X6 +X7 +X8 +X10	(P 41 , P215)
1 +X3 +X4 +X8 +X10	(P 43 , P439)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X9 +X10	(P 47 , P127)
1 +X2 +X4 +X6 +X8 +X9 +X10	(P 49 , P 29)
1 +X1 +X2 +X3 +X7 +X8 +X10	(P 53 , P 47)
1 +X3 +X4 +X5 +X8 +X9 +X10	(P 59 , P383)
1 +X1 +X4 +X5 +X6 +X7 +X8 +X9 +X10	(P 61 , P 7)
1 +X1 +X4 +X6 +X7 +X9 +X10	(P 71 , P 23)
1 +X1 +X2 +X6 +X8 +X9 +X10	(P 73 , P109)

1 +X1 +X2 +X5 +X6 +X7 +X10	(P 79 , P 5)
1 +X1 +X4 +X7 +X8 +X9 +X10	(P 83 , P 35)
1 +X1 +X2 +X6 +X7 +X8 +X10	(P 85 , P205)
1 +X1 +X2 +X4 +X6 +X7 +X10	(P 89 , P245)
1 +X2 +X4 +X5 +X7 +X9 +X10	(P 91 , P 25)
1 +X2 +X5 +X6 +X10	(P 95 , P 37)
1 +X2 +X3 +X5 +X10	(P101 , P347)
1 +X2 +X3 +X4 +X5 +X6 +X8 +X9 +X10	(P103 , P 91)
1 +X3 +X4 +X5 +X6 +X9 +X10	(P107 , P115)
1 +X1 +X2 +X5 +X10	(P109 , P251)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X8 +X10	(P115 , P157)
1 +X1 +X3 +X4 +X6 +X9 +X10	(P119 , P125)
1 +X6 +X7 +X9 +X10	(P125 , P 79)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X7 +X10	(P127 , P 71)
1 +X2 +X4 +X9 +X10	(P149 , P239)
1 +X5 +X8 +X9 +X10	(P151 , P 19)
1 +X1 +X3 +X5 +X6 +X8 +X10	(P157 , P223)
1 +X1 +X4 +X5 +X6 +X7 +X10	(P167 , P103)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X9 +X10	(P173 , P511)
1 +X2 +X3 +X7 +X8 +X9 +X10	(P175 , P 61)
1 +X3 +X7 +X9 +X10	(P179 , P191)
1 +X1 +X3 +X4 +X6 +X7 +X8 +X9 +X10	(P181 , P 1)

1 +X4 +X5 +X7 +X8 +X9 +X10	(P191 , P 89)
1 +X1 +X3 +X7 +X10	(P205 , P 13)
1 +X5 +X7 +X8 +X10	(P215 , P149)
1 +X3 +X4 +X6 +X8 +X9 +X10	(P221 , P 43)
1 +X2 +X5 +X9 +X10	(P223 , P173)
1 +X1 +X2 +X3 +X6 +X9 +X10	(P235 , P247)
1 +X1 +X2 +X4 +X6 +X8 +X10	(P239 , P 95)
1 +X2 +X6 +X7 +X10	(P245 , P 73)
1 +X1 +X6 +X9 +X10	(P247 , P107)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X9 +X10	(P251 , P175)
1 +X2 +X3 +X4 +X8 +X9 +X10	(P343 , P179)
1 +X1 +X6 +X8 +X10	(P347 , P 49)
1 +X2 +X3 +X4 +X5 +X8 +X10	(P367 , P101)
1 +X1 +X2 +X4 +X5 +X9 +X10	(P379 , P 83)
1 +X2 +X7 +X8 +X10	(P383 , P343)
1 +X1 +X2 +X4 +X8 +X9 +X10	(P439 , P151)
1 +X1 +X2 +X4 +X7 +X8 +X10	(P479 , P367)
1 +X7 +X10	(P511 , P479)

DEGREE 11

1 +X2 +X11	(P 1 , P 33)
1 +X2 +X5 +X8 +X11	(P 3 , P 99)
1 +X2 +X3 +X7 +X11	(P 5 , P 165)
1 +X2 +X3 +X5 +X11	(P 7 , P 231)
1 +X2 +X3 +X10+X11	(P 9 , P 293)
1 +X1 +X3 +X8 +X9 +X10+X11	(P 11 , P 363)
1 +X1 +X5 +X6 +X11	(P 13 , P 429)
1 +X1 +X4 +X5 +X6 +X8 +X11	(P 15 , P 495)
1 +X1 +X3 +X5 +X11	(P 17 , P 163)
1 +X1 +X4 +X9 +X11	(P 19 , P 423)
1 +X1 +X4 +X7 +X8 +X9 +X11	(P 21 , P 683)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X8 +X11	(P 25 , P 359)
1 +X1 +X3 +X4 +X7 +X10+X11	(P 27 , P 879)
1 +X1 +X3 +X4 +X5 +X7 +X8 +X10+X11	(P 29 , P 751)
1 +X1 +X2 +X3 +X4 +X7 +X9 +X10+X11	(P 31 , P1023)
1 +X2 +X3 +X4 +X6 +X7 +X9 +X10+X11	(P 33 , P 49)
1 +X2 +X6 +X8 +X11	(P 35 , P 57)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X9 +X11	(P 37 , P 179)
1 +X1 +X4 +X5 +X7 +X9 +X11	(P 39 , P 61)
1 +X4 +X5 +X6 +X7 +X9 +X11	(P 41 , P 309)
1 +X3 +X4 +X5 +X6 +X9 +X11	(P 43 , P 187)
1 +X1 +X2 +X3 +X4 +X7 +X8 +X10+X11	(P 45 , P 439)
1 +X1 +X3 +X4 +X5 +X6 +X9 +X10+X11	(P 47 , P 63)

1 +X3 +X6 +X7 +X8 +X9 +X11	(P 49 , P 229)
1 +X4 +X7 +X9 +X11	(P 51 , P 317)
1 +X1 +X2 +X6 +X7 +X10+X11	(P 53 , P 699)
1 +X3 +X7 +X10+X11	(P 55 , P 191)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X9 +X11	(P 57 , P 491)
1 +X1 +X3 +X4 +X6 +X8 +X11	(P 59 , P 447)
1 +X5 +X6 +X7 +X11	(P 61 , P 959)
1 +X1 +X2 +X4 +X7 +X8 +X9 +X10+X11	(P 63 , P 1)
1 +X3 +X6 +X7 +X8 +X10+X11	(P 67 , P 41)
1 +X2 +X3 +X6 +X7 +X8 +X9 +X10+X11	(P 71 , P 37)
1 +X1 +X5 +X6 +X7 +X10+X11	(P 73 , P 181)
1 +X1 +X2 +X4 +X7 +X10+X11	(P 75 , P 107)
1 +X1 +X4 +X5 +X7 +X10+X11	(P 77 , P 247)
1 +X2 +X3 +X4 +X7 +X9 +X11	(P 79 , P 35)
1 +X3 +X4 +X8 +X9 +X10+X11	(P 81 , P 295)
1 +X2 +X3 +X5 +X8 +X10+X11	(P 83 , P 173)
1 +X1 +X2 +X5 +X7 +X9 +X11	(P 85 , P 379)
1 +X2 +X4 +X5 +X7 +X9 +X11	(P 87 , P 103)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X8 +X11	(P 91 , P 239)
1 +X1 +X2 +X7 +X8 +X9 +X11	(P 93 , P 511)
1 +X4 +X5 +X6 +X11	(P 95 , P 17)
1 +X1 +X7 +X8 +X11	(P 99 , P 147)
1 +X4 +X5 +X6 +X8 +X10+X11	(P 101 , P 53)
1 +X1 +X2 +X6 +X9 +X10+X11	(P 103 , P 169)
1 +X5 +X9 +X10+X11	(P 105 , P 171)
1 +X3 +X5 +X7 +X11	(P 107 , P 371)

1 +X2 +X3 +X4 +X5 +X7 +X8 +X9 +X11	(P 109 , P 59)
1 +X1 +X3 +X4 +X5 +X6 +X11	(P 111 , P 101)
1 +X1 +X2 +X6 +X7 +X8 +X11	(P 113 , P 301)
1 +X1 +X4 +X5 +X8 +X9 +X11	(P 117 , P 183)
1 +X2 +X3 +X5 +X6 +X7 +X8 +X9 +X11	(P 119 , P 235)
1 +X5 +X6 +X7 +X8 +X10+X11	(P 121 , P 431)
1 +X2 +X4 +X7 +X8 +X9 +X11	(P 123 , P 503)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X10+X11	(P 125 , P 31)
1 +X3 +X5 +X6 +X7 +X10+X11	(P 127 , P 67)
1 +X2 +X3 +X4 +X5 +X7 +X8 +X10+X11	(P 137 , P 427)
1 +X2 +X3 +X5 +X7 +X8 +X9 +X10+X11	(P 139 , P 493)
1 +X3 +X4 +X6 +X8 +X9 +X11	(P 141 , P 189)
1 +X2 +X3 +X4 +X5 +X6 +X8 +X9 +X11	(P 143 , P 167)
1 +X1 +X5 +X7 +X9 +X10+X11	(P 147 , P 687)
1 +X4 +X7 +X8 +X9 +X10+X11	(P 149 , P 443)
1 +X4 +X5 +X6 +X9 +X10+X11	(P 151 , P 367)
1 +X3 +X4 +X6 +X7 +X8 +X11	(P 153 , P 887)
1 +X3 +X5 +X8 +X11	(P 155 , P 767)
1 +X1 +X2 +X3 +X5 +X6 +X8 +X10+X11	(P 157 , P 127)
1 +X3 +X5 +X7 +X8 +X10+X11	(P 159 , P 25)
1 +X2 +X5 +X6 +X7 +X8 +X9 +X10+X11	(P 163 , P 45)
1 +X1 +X2 +X3 +X6 +X7 +X9 +X10+X11	(P 165 , P 245)
1 +X2 +X7 +X9 +X11	(P 167 , P 155)
1 +X2 +X4 +X5 +X6 +X8 +X11	(P 169 , P 375)
1 +X2 +X4 +X5 +X6 +X7 +X8 +X10+X11	(P 171 , P 55)

1 +X2 +X3 +X4 +X6 +X8 +X9 +X10+X11	(P 173 , P 319)
1 +X2 +X3 +X6 +X7 +X10+X11	(P 175 , P 233)
1 +X1 +X3 +X5 +X7 +X8 +X11	(P 179 , P 175)
1 +X3 +X8 +X9 +X11	(P 181 , P 703)
1 +X2 +X5 +X6 +X8 +X9 +X11	(P 183 , P 415)
1 +X2 +X4 +X5 +X6 +X8 +X9 +X10+X11	(P 185 , P 895)
1 +X1 +X3 +X5 +X6 +X8 +X9 +X10+X11	(P 187 , P 15)
1 +X2 +X3 +X4 +X9 +X10+X11	(P 189 , P 3)
1 +X1 +X3 +X4 +X6 +X7 +X8 +X10+X11	(P 191 , P 81)
1 +X1 +X5 +X6 +X8 +X10+X11	(P 199 , P 213)
1 +X1 +X3 +X7 +X8 +X9 +X11	(P 201 , P 123)
1 +X1 +X3 +X10+X11	(P 203 , P 185)
1 +X1 +X2 +X5 +X7 +X8 +X9 +X10+X11	(P 205 , P 39)
1 +X1 +X2 +X6 +X8 +X10+X11	(P 211 , P 411)
1 +X1 +X2 +X3 +X4 +X10+X11	(P 213 , P 111)
1 +X1 +X4 +X5 +X6 +X7 +X9 +X10+X11	(P 215 , P 477)
1 +X7 +X9 +X10+X11	(P 217 , P 255)
1 +X1 +X3 +X4 +X5 +X7 +X9 +X10+X11	(P 219 , P 125)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X9 +X11	(P 221 , P 9)
1 +X2 +X3 +X4 +X5 +X6 +X9 +X10+X11	(P 223 , P 83)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X10+X11	(P 229 , P 139)
1 +X1 +X3 +X5 +X6 +X10+X11	(P 231 , P 343)
1 +X2 +X3 +X5 +X6 +X9 +X11	(P 233 , P 51)
1 +X2 +X4 +X9 +X11	(P 235 , P 315)
1 +X1 +X2 +X5 +X6 +X7 +X11	(P 237 , P 105)
1 +X3 +X4 +X5 +X9 +X10+X11	(P 239 , P 365)

1 +X1 +X2 +X5 +X9 +X10+X11	(P 243 , P 695)
1 +X2 +X4 +X6 +X7 +X9 +X11	(P 245 , P 243)
1 +X1 +X2 +X3 +X4 +X5 +X8 +X9 +X11	(P 247 , P 763)
1 +X1 +X4 +X8 +X11	(P 249 , P 29)
1 +X2 +X4 +X6 +X7 +X10+X11	(P 251 , P 95)
1 +X6 +X8 +X9 +X11	(P 255 , P 199)
1 +X1 +X4 +X6 +X7 +X8 +X9 +X10+X11	(P 293 , P 311)
1 +X1 +X2 +X4 +X6 +X7 +X8 +X10+X11	(P 295 , P 47)
1 +X1 +X2 +X9 +X11	(P 301 , P 237)
1 +X2 +X3 +X4 +X6 +X7 +X8 +X9 +X11	(P 303 , P 159)
1 +X2 +X3 +X5 +X6 +X8 +X9 +X10+X11	(P 307 , P 383)
1 +X4 +X5 +X7 +X11	(P 309 , P 507)
1 +X1 +X2 +X4 +X11	(P 311 , P 7)
1 +X3 +X6 +X8 +X11	(P 315 , P 5)
1 +X3 +X4 +X5 +X6 +X7 +X9 +X10+X11	(P 317 , P 113)
1 +X1 +X4 +X7 +X8 +X10+X11	(P 319 , P 73)
1 +X1 +X2 +X5 +X8 +X10+X11	(P 331 , P 43)
1 +X5 +X6 +X9 +X11	(P 333 , P 303)
1 +X4 +X6 +X8 +X11	(P 335 , P 205)
1 +X4 +X5 +X8 +X9 +X10+X11	(P 339 , P 119)
1 +X1 +X3 +X5 +X7 +X9 +X11	(P 341 , P 509)
1 +X1 +X7 +X8 +X9 +X10+X11	(P 343 , P 121)
1 +X4 +X6 +X7 +X11	(P 347 , P 19)
1 +X2 +X3 +X8 +X11	(P 349 , P 21)
1 +X1 +X4 +X5 +X9 +X10+X11	(P 351 , P 149)

1 +X1 +X3 +X5 +X9 +X10+X11	(P 359 , P 307)
1 +X2 +X5 +X6 +X11	(P 363 , P 109)
1 +X2 +X9 +X10+X11	(P 365 , P 151)
1 +X1 +X2 +X6 +X11	(P 367 , P 469)
1 +X1 +X2 +X3 +X4 +X6 +X9 +X10+X11	(P 371 , P 251)
1 +X1 +X2 +X3 +X5 +X7 +X8 +X9 +X11	(P 373 , P 27)
1 +X2 +X3 +X5 +X7 +X8 +X11	(P 375 , P 93)
1 +X1 +X4 +X6 +X7 +X10+X11	(P 379 , P 71)
1 +X1 +X3 +X4 +X7 +X8 +X9 +X10+X11	(P 381 , P 201)
1 +X5 +X6 +X10+X11	(P 383 , P 331)
1 +X1 +X2 +X3 +X5 +X6 +X8 +X9 +X11	(P 411 , P 13)
1 +X3 +X4 +X6 +X8 +X10+X11	(P 413 , P 117)
1 +X2 +X4 +X7 +X11	(P 415 , P 91)
1 +X1 +X8 +X10+X11	(P 423 , P 221)
1 +X1 +X2 +X3 +X6 +X7 +X11	(P 427 , P 143)
1 +X1 +X3 +X6 +X9 +X10+X11	(P 429 , P 501)
1 +X1 +X3 +X5 +X6 +X7 +X11	(P 431 , P 351)
1 +X2 +X3 +X4 +X8 +X10+X11	(P 439 , P 79)
1 +X3 +X4 +X5 +X7 +X8 +X11	(P 443 , P 137)
1 +X3 +X5 +X6 +X7 +X8 +X9 +X10+X11	(P 447 , P 211)
1 +X3 +X4 +X10+X11	(P 463 , P 475)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X9 +X11	(P 469 , P 249)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X9 +X11	(P 471 , P 381)
1 +X1 +X2 +X4 +X6 +X10+X11	(P 475 , P 85)
1 +X1 +X2 +X3 +X4 +X6 +X8 +X9 +X11	(P 477 , P 75)
1 +X2 +X3 +X4 +X5 +X8 +X11	(P 479 , P 215)

1 +X1 +X3 +X6 +X8 +X9 +X11	(P 491 , P 373)
1 +X1 +X4 +X7 +X9 +X10+X11	(P 493 , P 335)
1 +X1 +X3 +X4 +X5 +X8 +X11	(P 495 , P 735)
1 +X2 +X5 +X6 +X7 +X8 +X11	(P 501 , P 157)
1 +X3 +X5 +X9 +X11	(P 503 , P 223)
1 +X2 +X7 +X10+X11	(P 507 , P 203)
1 +X1 +X2 +X3 +X8 +X10+X11	(P 509 , P 333)
1 +X3 +X6 +X9 +X11	(P 511 , P 463)
1 +X2 +X4 +X6 +X8 +X10+X11	(P 683 , P 11)
1 +X2 +X4 +X6 +X9 +X10+X11	(P 687 , P 77)
1 +X3 +X5 +X6 +X7 +X9 +X11	(P 695 , P 141)
1 +X1 +X2 +X3 +X4 +X7 +X11	(P 699 , P 153)
1 +X2 +X3 +X4 +X7 +X10+X11	(P 703 , P 341)
1 +X1 +X2 +X4 +X5 +X8 +X9 +X10+X11	(P 727 , P 87)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X10+X11	(P 731 , P 217)
1 +X2 +X4 +X5 +X6 +X7 +X11	(P 735 , P 347)
1 +X1 +X2 +X3 +X7 +X8 +X11	(P 751 , P 219)
1 +X2 +X4 +X5 +X7 +X8 +X9 +X10+X11	(P 763 , P 413)
1 +X4 +X8 +X9 +X11	(P 767 , P 727)
1 +X1 +X4 +X5 +X6 +X10+X11	(P 879 , P 349)
1 +X1 +X3 +X4 +X6 +X7 +X8 +X9 +X11	(P 887 , P 339)
1 +X1 +X8 +X9 +X11	(P 895 , P 731)
1 +X6 +X8 +X10+X11	(P 959 , P 471)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X9 +X11	(P 991 , P 479)
1 +X9 +X11	(P1023 , P 991)

DEGREE 12

1 +X1 +X4 +X6 +X12	(P 1 , P 127)
1 +X1 +X2 +X5 +X7 +X8 +X9 +X11+X12	(P 11 , P1367)
1 +X1 +X3 +X4 +X6 +X8 +X10+X11+X12	(P 17 , P 223)
1 +X1 +X2 +X5 +X10+X11+X12	(P 19 , P 731)
1 +X2 +X3 +X9 +X12	(P 23 , P 877)
1 +X1 +X2 +X4 +X6 +X11+X12	(P 29 , P 499)
1 +X1 +X3 +X4 +X5 +X7 +X8 +X9 +X10+X11+X12	(P 31 , P 251)
1 +X2 +X4 +X5 +X6 +X8 +X9 +X10+X12	(P 37 , P 151)
1 +X5 +X6 +X7 +X9 +X11+X12	(P 41 , P 139)
1 +X1 +X3 +X5 +X9 +X11+X12	(P 43 , P 683)
1 +X4 +X7 +X8 +X9 +X11+X12	(P 47 , P 629)
1 +X4 +X6 +X7 +X9 +X11+X12	(P 53 , P 613)
1 +X1 +X2 +X3 +X8 +X9 +X12	(P 59 , P 437)
1 +X2 +X6 +X8 +X9 +X10+X12	(P 61 , P 313)
1 +X2 +X3 +X4 +X5 +X7 +X9 +X10+X12	(P 67 , P 319)
1 +X3 +X4 +X5 +X8 +X9 +X12	(P 71 , P 827)
1 +X1 +X2 +X3 +X4 +X7 +X10+X11+X12	(P 73 , P 229)
1 +X2 +X3 +X6 +X8 +X10+X12	(P 79 , P 823)
1 +X2 +X3 +X5 +X7 +X10+X12	(P 83 , P 607)
1 +X3 +X4 +X5 +X7 +X9 +X12	(P 89 , P 167)
1 +X4 +X6 +X9 +X10+X11+X12	(P 97 , P 17)
1 +X1 +X2 +X4 +X7 +X11+X12	(P 101 , P 241)

1 +X1 +X2 +X3 +X6 +X10+X12	(P 103 , P 199)
1 +X2 +X3 +X4 +X6 +X11+X12	(P 107 , P 163)
1 +X3 +X6 +X7 +X8 +X11+X12	(P 109 , P 179)
1 +X3 +X4 +X6 +X9 +X10+X12	(P 113 , P 37)
1 +X4 +X6 +X8 +X10+X11+X12	(P 121 , P 43)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X8 +X9 +X11+X12	(P 127 , P 79)
1 +X1 +X2 +X3 +X4 +X8 +X9 +X11+X12	(P 137 , P1019)
1 +X1 +X2 +X4 +X5 +X10+X12	(P 139 , P 671)
1 +X1 +X2 +X5 +X6 +X10+X12	(P 149 , P 991)
1 +X1 +X2 +X3 +X6 +X7 +X8 +X11+X12	(P 151 , P1499)
1 +X2 +X3 +X4 +X5 +X7 +X8 +X11+X12	(P 157 , P1007)
1 +X1 +X4 +X5 +X7 +X8 +X12	(P 163 , P 113)
1 +X2 +X3 +X6 +X10+X11+X12	(P 167 , P 367)
1 +X1 +X2 +X5 +X7 +X10+X12	(P 173 , P 187)
1 +X2 +X3 +X4 +X5 +X7 +X8 +X9 +X10+X11+X12	(P 179 , P 421)
1 +X3 +X5 +X6 +X12	(P 181 , P 157)
1 +X1 +X3 +X7 +X8 +X11+X12	(P 187 , P 691)
1 +X2 +X3 +X4 +X6 +X10+X12	(P 191 , P 443)
1 +X5 +X8 +X9 +X12	(P 197 , P 71)
1 +X4 +X6 +X7 +X12	(P 199 , P 703)
1 +X1 +X2 +X4 +X5 +X9 +X12	(P 209 , P1403)
1 +X1 +X3 +X4 +X8 +X11+X12	(P 211 , P 359)
1 +X1 +X2 +X4 +X6 +X8 +X12	(P 223 , P1013)
1 +X1 +X3 +X5 +X6 +X7 +X8 +X10+X12	(P 227 , P 41)
1 +X3 +X4 +X8 +X9 +X10+X12	(P 229 , P 209)

1 +X1 +X3 +X6 +X7 +X9 +X12	(P 233 , P 463)
1 +X2 +X4 +X8 +X9 +X10+X12	(P 239 , P 211)
1 +X2 +X3 +X4 +X8 +X10+X12	(P 241 , P 719)
1 +X2 +X4 +X6 +X9 +X10+X12	(P 251 , P 409)
1 +X1 +X3 +X4 +X5 +X8 +X12	(P 253 , P 347)
1 +X4 +X10+X11+X12	(P 277 , P 743)
1 +X2 +X4 +X9 +X10+X11+X12	(P 281 , P1759)
1 +X1 +X4 +X6 +X7 +X8 +X9 +X11+X12	(P 283 , P 439)
1 +X1 +X2 +X3 +X4 +X5 +X8 +X9 +X12	(P 293 , P 89)
1 +X4 +X7 +X11+X12	(P 307 , P 173)
1 +X2 +X6 +X7 +X8 +X10+X12	(P 311 , P 661)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X11+X12	(P 313 , P 181)
1 +X1 +X4 +X5 +X9 +X11+X12	(P 317 , P 821)
1 +X3 +X4 +X9 +X10+X11+X12	(P 319 , P 599)
1 +X1 +X2 +X8 +X10+X11+X12	(P 331 , P 253)
1 +X4 +X5 +X6 +X8 +X9 +X12	(P 341 , P 619)
1 +X3 +X6 +X7 +X8 +X9 +X10+X11+X12	(P 347 , P 191)
1 +X6 +X7 +X8 +X9 +X10+X12	(P 349 , P 733)
1 +X2 +X3 +X4 +X5 +X9 +X12	(P 359 , P 137)
1 +X1 +X4 +X5 +X6 +X7 +X10+X11+X12	(P 361 , P 281)
1 +X3 +X5 +X6 +X9 +X11+X12	(P 367 , P 227)
1 +X2 +X3 +X4 +X5 +X6 +X12	(P 373 , P 361)
1 +X1 +X4 +X5 +X7 +X8 +X9 +X10+X12	(P 379 , P 67)
1 +X1 +X6 +X8 +X10+X11+X12	(P 383 , P 103)

1 +X1 +X2 +X4 +X6 +X7 +X8 +X10+X12	(P 397 , P1021)
1 +X1 +X2 +X3 +X5 +X9 +X10+X11+X12	(P 407 , P1003)
1 +X1 +X2 +X3 +X4 +X8 +X12	(P 409 , P 943)
1 +X3 +X4 +X5 +X7 +X11+X12	(P 421 , P 29)
1 +X2 +X3 +X4 +X8 +X9 +X12	(P 431 , P 751)
1 +X1 +X4 +X7 +X8 +X10+X12	(P 437 , P 283)
1 +X1 +X2 +X10+X12	(P 439 , P 941)
1 +X1 +X5 +X6 +X7 +X9 +X10+X11+X12	(P 443 , P 701)
1 +X2 +X10+X11+X12	(P 457 , P 331)
1 +X1 +X2 +X3 +X7 +X9 +X10+X11+X12	(P 461 , P 83)
1 +X2 +X4 +X5 +X6 +X7 +X9 +X11+X12	(P 463 , P1471)
1 +X3 +X7 +X8 +X9 +X10+X12	(P 467 , P1783)
1 +X2 +X4 +X5 +X6 +X10+X12	(P 473 , P1387)
1 +X2 +X3 +X6 +X8 +X9 +X12	(P 479 , P1531)
1 +X5 +X6 +X8 +X12	(P 487 , P 53)
1 +X1 +X2 +X6 +X9 +X10+X12	(P 491 , P 233)
1 +X1 +X4 +X5 +X6 +X9 +X10+X11+X12	(P 493 , P 293)
1 +X2 +X6 +X9 +X10+X11+X12	(P 499 , P 487)
1 +X3 +X4 +X7 +X8 +X9 +X12	(P 503 , P 307)
1 +X3 +X9 +X10+X12	(P 509 , P 587)
1 +X2 +X5 +X6 +X8 +X11+X12	(P 587 , P 461)
1 +X3 +X6 +X7 +X10+X11+X12	(P 589 , P 277)
1 +X2 +X4 +X5 +X8 +X11+X12	(P 599 , P 631)
1 +X1 +X4 +X5 +X6 +X9 +X12	(P 607 , P 829)

1 +X1 +X3 +X8 +X9 +X11+X12	(P 613 , P 23)
1 +X1 +X2 +X5 +X6 +X9 +X10+X11+X12	(P 619 , P 101)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X9 +X12	(P 629 , P 61)
1 +X1 +X3 +X4 +X5 +X6 +X8 +X11+X12	(P 631 , P 457)
1 +X2 +X7 +X8 +X12	(P 661 , P2047)
1 +X1 +X2 +X3 +X6 +X7 +X10+X11+X12	(P 667 , P 863)
1 +X1 +X6 +X8 +X9 +X10+X12	(P 671 , P 983)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X8 +X12	(P 683 , P 373)
1 +X6 +X7 +X8 +X9 +X11+X12	(P 691 , P 311)
1 +X6 +X7 +X9 +X12	(P 701 , P 379)
1 +X1 +X3 +X5 +X6 +X8 +X12	(P 703 , P 859)
1 +X1 +X4 +X8 +X9 +X11+X12	(P 719 , P 467)
1 +X1 +X5 +X7 +X8 +X9 +X12	(P 727 , P 383)
1 +X1 +X2 +X5 +X6 +X9 +X12	(P 731 , P1399)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X11+X12	(P 733 , P 887)
1 +X2 +X4 +X5 +X6 +X8 +X10+X11+X12	(P 743 , P 11)
1 +X3 +X7 +X8 +X10+X11+X12	(P 751 , P 149)
1 +X2 +X5 +X7 +X10+X11+X12	(P 757 , P 317)
1 +X1 +X3 +X4 +X5 +X6 +X12	(P 821 , P 473)
1 +X4 +X8 +X9 +X10+X11+X12	(P 823 , P 197)
1 +X1 +X5 +X8 +X12	(P 827 , P 757)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X8 +X9 +X10+X12	(P 829 , P 727)
1 +X4 +X5 +X6 +X8 +X9 +X10+X11+X12	(P 853 , P 349)
1 +X1 +X3 +X4 +X9 +X11+X12	(P 859 , P 509)
1 +X2 +X5 +X8 +X10+X11+X12	(P 863 , P 239)

1 +X1 +X4 +X6 +X7 +X10+X12	(P 877 , P 407)
1 +X1 +X2 +X3 +X8 +X10+X12	(P 887 , P 73)
1 +X3 +X5 +X7 +X8 +X9 +X12	(P 893 , P 491)
1 +X1 +X2 +X4 +X10+X11+X12	(P 941 , P 47)
1 +X3 +X4 +X7 +X12	(P 943 , P 503)
1 +X4 +X5 +X7 +X8 +X11+X12	(P 983 , P 479)
1 +X2 +X7 +X8 +X10+X11+X12	(P 989 , P 107)
1 +X1 +X2 +X3 +X6 +X8 +X12	(P 991 , P1919)
1 +X2 +X5 +X7 +X9 +X10+X12	(P1003 , P 109)
1 +X2 +X3 +X5 +X7 +X8 +X9 +X10+X12	(P1007 , P 59)
1 +X1 +X3 +X7 +X9 +X11+X12	(P1013 , P 853)
1 +X1 +X2 +X7 +X10+X11+X12	(P1019 , P 589)
1 +X1 +X3 +X4 +X5 +X7 +X10+X11+X12	(P1021 , P 341)
1 +X3 +X4 +X6 +X7 +X8 +X12	(P1367 , P 667)
1 +X4 +X5 +X10+X12	(P1387 , P 1)
1 +X1 +X2 +X8 +X12	(P1399 , P 397)
1 +X2 +X6 +X7 +X10+X11+X12	(P1403 , P 97)
1 +X1 +X3 +X5 +X6 +X7 +X12	(P1471 , P 989)
1 +X3 +X4 +X7 +X8 +X9 +X10+X11+X12	(P1499 , P 893)
1 +X2 +X3 +X4 +X6 +X7 +X8 +X10+X12	(P1531 , P 493)
1 +X1 +X2 +X5 +X8 +X9 +X10+X11+X12	(P1759 , P 431)
1 +X1 +X3 +X4 +X8 +X9 +X10+X11+X12	(P1783 , P 19)
1 +X1 +X2 +X4 +X6 +X8 +X9 +X11+X12	(P1919 , P 121)
1 +X6 +X8 +X11+X12	(P2047 , P 31)

DEGREE 13

1 +X1 +X3 +X4 +X13	(P 1 , P 65)
1 +X4 +X5 +X7 +X9 +X10+X13	(P 3 , P 195)
1 +X1 +X4 +X7 +X8 +X11+X13	(P 5 , P 325)
1 +X1 +X2 +X3 +X6 +X8 +X9 +X10+X13	(P 7 , P 455)
1 +X5 +X6 +X7 +X8 +X12+X13	(P 9 , P 585)
1 +X1 +X5 +X7 +X8 +X9 +X13	(P 11 , P 715)
1 +X3 +X4 +X5 +X6 +X12+X13	(P 13 , P 845)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X9 +X13	(P 15 , P 975)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X11+X13	(P 17 , P 581)
1 +X3 +X5 +X9 +X11+X12+X13	(P 19 , P1235)
1 +X1 +X4 +X6 +X7 +X8 +X11+X12+X13	(P 21 , P1365)
1 +X1 +X2 +X5 +X11+X12+X13	(P 23 , P1495)
1 +X2 +X3 +X4 +X6 +X8 +X10+X12+X13	(P 25 , P1227)
1 +X2 +X4 +X8 +X9 +X12+X13	(P 27 , P1755)
1 +X2 +X6 +X8 +X9 +X10+X11+X12+X13	(P 29 , P1885)
1 +X2 +X3 +X6 +X8 +X10+X11+X12+X13	(P 31 , P2015)
1 +X1 +X2 +X3 +X4 +X5 +X10+X11+X13	(P 33 , P 323)
1 +X1 +X3 +X4 +X5 +X7 +X8 +X9 +X11+X12+X13	(P 35 , P 839)
1 +X2 +X3 +X4 +X7 +X8 +X13	(P 37 , P1355)
1 +X1 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X12+X13	(P 39 , P1871)
1 +X2 +X5 +X6 +X7 +X8 +X9 +X11+X13	(P 41 , P1357)
1 +X2 +X3 +X4 +X5 +X6 +X8 +X10+X11+X12+X13	(P 43 , P2795)
1 +X1 +X3 +X5 +X7 +X8 +X10+X11+X13	(P 45 , P2779)

1 +X1 +X3 +X4 +X6 +X9 +X13	(P 47 , P3055)
1 +X5 +X8 +X10+X13	(P 49 , P 711)
1 +X1 +X2 +X3 +X8 +X12+X13	(P 51 , P1743)
1 +X1 +X2 +X3 +X6 +X8 +X10+X12+X13	(P 53 , P2775)
1 +X2 +X3 +X4 +X6 +X7 +X9 +X10+X11+X12+X13	(P 55 , P3575)
1 +X1 +X2 +X4 +X6 +X7 +X9 +X11+X13	(P 57 , P1487)
1 +X3 +X4 +X7 +X9 +X10+X13	(P 59 , P3551)
1 +X3 +X6 +X8 +X9 +X11+X13	(P 61 , P3039)
1 +X1 +X3 +X4 +X6 +X8 +X10+X11+X13	(P 63 , P4095)
1 +X1 +X5 +X6 +X7 +X9 +X10+X12+X13	(P 65 , P 97)
1 +X1 +X2 +X4 +X6 +X7 +X8 +X9 +X10+X12+X13	(P 67 , P 113)
1 +X3 +X4 +X5 +X7 +X9 +X10+X11+X13	(P 69 , P 355)
1 +X1 +X2 +X3 +X9 +X11+X13	(P 71 , P 121)
1 +X5 +X10+X11+X13	(P 73 , P 613)
1 +X1 +X6 +X9 +X13	(P 75 , P 371)
1 +X1 +X4 +X5 +X7 +X9 +X10+X11+X13	(P 77 , P 871)
1 +X1 +X3 +X6 +X8 +X11+X13	(P 79 , P 125)
1 +X3 +X6 +X7 +X10+X12+X13	(P 81 , P 841)
1 +X1 +X5 +X6 +X7 +X8 +X9 +X12+X13	(P 83 , P 629)
1 +X1 +X2 +X3 +X4 +X11+X13	(P 85 , P1387)
1 +X3 +X6 +X7 +X8 +X12+X13	(P 87 , P 379)
1 +X5 +X7 +X8 +X10+X12+X13	(P 89 , P1645)
1 +X1 +X2 +X3 +X5 +X7 +X8 +X11+X13	(P 91 , P 887)
1 +X1 +X2 +X3 +X4 +X8 +X10+X12+X13	(P 93 , P1903)
1 +X1 +X3 +X7 +X13	(P 95 , P 127)

1 +X1 +X3 +X4 +X7 +X8 +X9 +X11+X13	(P 97 , P 453)
1 +X1 +X6 +X7 +X9 +X12+X13	(P 99 , P 969)
1 +X2 +X4 +X6 +X8 +X10+X13	(P 101 , P1485)
1 +X1 +X2 +X4 +X7 +X8 +X11+X12+X13	(P 103 , P 637)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X11+X13	(P 105 , P1877)
1 +X1 +X2 +X7 +X8 +X9 +X13	(P 107 , P1403)
1 +X1 +X4 +X5 +X6 +X7 +X9 +X11+X13	(P 109 , P2935)
1 +X1 +X2 +X3 +X7 +X9 +X10+X11+X13	(P 111 , P 383)
1 +X5 +X6 +X7 +X8 +X9 +X10+X12+X13	(P 113 , P 971)
1 +X3 +X7 +X8 +X9 +X10+X11+X12+X13	(P 115 , P1661)
1 +X1 +X2 +X3 +X4 +X5 +X9 +X10+X13	(P 117 , P3035)
1 +X1 +X4 +X5 +X7 +X9 +X13	(P 119 , P 895)
1 +X3 +X9 +X12+X13	(P 121 , P2007)
1 +X3 +X5 +X9 +X10+X11+X13	(P 123 , P1919)
1 +X1 +X2 +X3 +X4 +X5 +X8 +X9 +X11+X12+X13	(P 125 , P3967)
1 +X2 +X3 +X5 +X7 +X9 +X10+X12+X13	(P 127 , P 1)
1 +X3 +X5 +X6 +X11+X12+X13	(P 131 , P 81)
1 +X2 +X8 +X9 +X10+X11+X13	(P 133 , P 227)
1 +X5 +X7 +X8 +X11+X12+X13	(P 135 , P 73)
1 +X2 +X5 +X8 +X10+X12+X13	(P 137 , P 357)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X10+X11+X12+X13	(P 139 , P 211)
1 +X2 +X3 +X4 +X5 +X7 +X8 +X10+X13	(P 141 , P 487)
1 +X1 +X3 +X4 +X5 +X10+X11+X12+X13	(P 143 , P 69)
1 +X4 +X6 +X9 +X11+X12+X13	(P 145 , P 617)
1 +X6 +X8 +X10+X11+X12+X13	(P 147 , P 341)
1 +X1 +X2 +X3 +X5 +X9 +X10+X12+X13	(P 149 , P 747)

1 +X1 +X8 +X10+X11+X12+X13	(P 151 , P 203)
1 +X1 +X2 +X4 +X5 +X6 +X8 +X9 +X11+X12+X13	(P 153 , P 877)
1 +X1 +X2 +X4 +X5 +X6 +X9 +X10+X13	(P 155 , P 471)
1 +X1 +X2 +X3 +X7 +X10+X11+X12+X13	(P 157 , P1007)
1 +X1 +X3 +X4 +X7 +X10+X13	(P 159 , P 67)
1 +X1 +X3 +X4 +X6 +X7 +X10+X12+X13	(P 161 , P 593)
1 +X1 +X6 +X7 +X8 +X11+X13	(P 163 , P 601)
1 +X1 +X4 +X5 +X6 +X9 +X10+X12+X13	(P 165 , P1267)
1 +X1 +X2 +X3 +X5 +X6 +X8 +X8 +X10+X12+X13	(P 167 , P 333)
1 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X13	(P 169 , P1397)
1 +X3 +X4 +X11+X13	(P 171 , P 731)
1 +X2 +X4 +X12+X13	(P 173 , P1527)
1 +X2 +X5 +X6 +X9 +X10+X11+X12+X13	(P 175 , P 199)
1 +X1 +X2 +X4 +X6 +X7 +X9 +X12+X13	(P 177 , P1231)
1 +X4 +X7 +X9 +X10+X11+X13	(P 179 , P 861)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X8 +X9 +X11+X13	(P 181 , P1787)
1 +X1 +X7 +X10+X13	(P 183 , P 463)
1 +X1 +X2 +X3 +X4 +X8 +X9 +X10+X11+X12+X13	(P 185 , P1917)
1 +X1 +X2 +X3 +X4 +X6 +X8 +X9 +X10+X11+X13	(P 187 , P 991)
1 +X2 +X3 +X4 +X9 +X12+X13	(P 189 , P2047)
1 +X1 +X2 +X4 +X5 +X8 +X9 +X10+X11+X12+X13	(P 191 , P 33)
1 +X1 +X4 +X5 +X6 +X7 +X8 +X11+X13	(P 195 , P 291)
1 +X2 +X5 +X7 +X13	(P 197 , P 105)
1 +X1 +X6 +X8 +X13	(P 199 , P 549)
1 +X4 +X5 +X6 +X7 +X8 +X13	(P 201 , P 339)

1 +X2 +X3 +X5 +X6 +X8 +X9 +X11+X13	(P 203 , P 807)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X12+X13	(P 205 , P 117)
1 +X1 +X2 +X3 +X4 +X9 +X10+X12+X13	(P 207 , P 329)
1 +X8 +X11+X12+X13	(P 209 , P 597)
1 +X2 +X3 +X6 +X7 +X8 +X10+X12+X13	(P 211 , P1323)
1 +X1 +X2 +X5 +X8 +X9 +X13	(P 213 , P 363)
1 +X4 +X8 +X9 +X10+X11+X13	(P 215 , P 723)
1 +X1 +X4 +X5 +X6 +X7 +X13	(P 217 , P 855)
1 +X6 +X8 +X9 +X10+X12+X13	(P 219 , P1511)
1 +X2 +X4 +X8 +X13	(P 221 , P 123)
1 +X1 +X2 +X5 +X6 +X7 +X10+X12+X13	(P 223 , P 197)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X8 +X13	(P 225 , P 713)
1 +X1 +X2 +X6 +X7 +X10+X13	(P 227 , P1229)
1 +X5 +X8 +X9 +X10+X12+X13	(P 229 , P 621)
1 +X1 +X5 +X7 +X8 +X9 +X10+X12+X13	(P 231 , P 853)
1 +X2 +X4 +X5 +X6 +X11+X13	(P 233 , P1371)
1 +X1 +X2 +X4 +X8 +X9 +X10+X11+X13	(P 235 , P1771)
1 +X7 +X8 +X11+X13	(P 237 , P 375)
1 +X4 +X6 +X7 +X8 +X10+X11+X12+X13	(P 239 , P 459)
1 +X1 +X4 +X5 +X6 +X7 +X11+X12+X13	(P 241 , P1491)
1 +X1 +X3 +X4 +X5 +X7 +X8 +X9 +X13	(P 243 , P1901)
1 +X1 +X2 +X4 +X5 +X6 +X10+X12+X13	(P 245 , P 879)
1 +X4 +X5 +X7 +X8 +X9 +X13	(P 247 , P 983)
1 +X1 +X3 +X6 +X7 +X8 +X11+X12+X13	(P 249 , P1887)
1 +X3 +X6 +X9 +X10+X12+X13	(P 251 , P2031)

1 +X6 +X10+X12+X13	(P 253 , P 63)
1 +X1 +X2 +X3 +X5 +X7 +X10+X11+X13	(P 255 , P 131)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X9 +X10+X12+X13	(P 265 , P 843)
1 +X5 +X6 +X7 +X10+X11+X13	(P 267 , P 973)
1 +X1 +X3 +X5 +X7 +X8 +X9 +X12+X13	(P 269 , P 633)
1 +X3 +X6 +X8 +X9 +X10+X11+X12+X13	(P 271 , P 589)
1 +X1 +X9 +X10+X13	(P 273 , P1363)
1 +X1 +X2 +X3 +X5 +X7 +X8 +X10+X13	(P 275 , P1493)
1 +X2 +X3 +X4 +X9 +X11+X13	(P 277 , P1395)
1 +X1 +X2 +X4 +X5 +X7 +X9 +X10+X13	(P 279 , P1243)
1 +X2 +X11+X12+X13	(P 281 , P1883)
1 +X2 +X3 +X5 +X6 +X8 +X11+X12+X13	(P 283 , P2013)
1 +X2 +X7 +X11+X13	(P 285 , P 381)
1 +X4 +X5 +X6 +X8 +X9 +X13	(P 287 , P 327)
1 +X3 +X7 +X8 +X10+X11+X13	(P 291 , P1359)
1 +X3 +X4 +X5 +X7 +X8 +X10+X12+X13	(P 293 , P1653)
1 +X2 +X5 +X7 +X9 +X11+X13	(P 295 , P1373)
1 +X1 +X2 +X8 +X9 +X12+X13	(P 297 , P2907)
1 +X3 +X4 +X5 +X8 +X9 +X10+X12+X13	(P 299 , P2811)
1 +X1 +X3 +X7 +X8 +X10+X13	(P 301 , P 891)
1 +X2 +X5 +X7 +X8 +X9 +X11+X12+X13	(P 303 , P 719)
1 +X3 +X4 +X5 +X6 +X7 +X10+X12+X13	(P 305 , P1751)
1 +X1 +X2 +X3 +X6 +X8 +X11+X12+X13	(P 307 , P2783)
1 +X2 +X4 +X7 +X10+X11+X13	(P 309 , P1911)
1 +X4 +X5 +X6 +X8 +X9 +X11+X12+X13	(P 311 , P1503)

1 +X1 +X4 +X5 +X7 +X8 +X9 +X11+X13	(P 313 , P3567)
1 +X1 +X5 +X11+X13	(P 315 , P3071)
1 +X1 +X5 +X7 +X8 +X12+X13	(P 317 , P 255)
1 +X2 +X3 +X4 +X8 +X10+X13	(P 319 , P 49)
1 +X1 +X2 +X3 +X5 +X6 +X13	(P 323 , P 89)
1 +X3 +X5 +X6 +X9 +X10+X11+X12+X13	(P 325 , P 485)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X11+X13	(P 327 , P 307)
1 +X2 +X4 +X6 +X7 +X9 +X10+X12+X13	(P 329 , P 743)
1 +X1 +X2 +X4 +X5 +X8 +X10+X12+X13	(P 331 , P 109)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X12+X13	(P 333 , P1001)
1 +X3 +X4 +X7 +X8 +X10+X13	(P 335 , P 565)
1 +X1 +X2 +X3 +X7 +X8 +X10+X11+X13	(P 337 , P1259)
1 +X2 +X3 +X4 +X5 +X7 +X13	(P 339 , P 347)
1 +X2 +X4 +X7 +X8 +X10+X11+X12+X13	(P 341 , P1517)
1 +X5 +X6 +X7 +X13	(P 343 , P 823)
1 +X1 +X2 +X4 +X6 +X8 +X9 +X10+X11+X12+X13	(P 345 , P1775)
1 +X1 +X3 +X4 +X6 +X7 +X9 +X12+X13	(P 347 , P 119)
1 +X3 +X4 +X5 +X9 +X12+X13	(P 349 , P 639)
1 +X1 +X3 +X7 +X8 +X9 +X11+X12+X13	(P 351 , P 457)
1 +X1 +X3 +X5 +X6 +X7 +X9 +X10+X13	(P 355 , P 605)
1 +X1 +X5 +X7 +X10+X11+X13	(P 357 , P2005)
1 +X1 +X3 +X5 +X8 +X9 +X13	(P 359 , P1339)
1 +X2 +X3 +X10+X11+X12+X13	(P 361 , P2807)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X10+X11+X12+X13	(P 363 , P 367)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X8 +X11+X12+X13	(P 365 , P1407)

1 +X5 +X6 +X7 +X11+X12+X13	(P 367 , P 979)
1 +X1 +X2 +X5 +X6 +X8 +X11+X12+X13	(P 369 , P2011)
1 +X2 +X4 +X5 +X8 +X9 +X11+X12+X13	(P 371 , P 863)
1 +X4 +X8 +X11+X13	(P 373 , P2943)
1 +X1 +X4 +X5 +X6 +X8 +X9 +X10+X13	(P 375 , P1855)
1 +X1 +X2 +X3 +X4 +X10+X11+X12+X13	(P 377 , P3839)
1 +X1 +X5 +X6 +X8 +X12+X13	(P 379 , P 31)
1 +X1 +X4 +X9 +X10+X11+X13	(P 381 , P 3)
1 +X2 +X4 +X5 +X7 +X10+X13	(P 383 , P 161)
1 +X1 +X2 +X4 +X5 +X7 +X9 +X12+X13	(P 391 , P 421)
1 +X1 +X3 +X4 +X6 +X9 +X10+X11+X13	(P 393 , P 243)
1 +X2 +X6 +X7 +X8 +X12+X13	(P 395 , P 551)
1 +X2 +X4 +X5 +X8 +X12+X13	(P 397 , P 77)
1 +X2 +X3 +X6 +X7 +X9 +X11+X12+X13	(P 399 , P 681)
1 +X3 +X4 +X7 +X10+X12+X13	(P 401 , P 373)
1 +X1 +X2 +X7 +X10+X12+X13	(P 403 , P 811)
1 +X6 +X10+X11+X13	(P 405 , P 219)
1 +X1 +X2 +X4 +X5 +X6 +X8 +X10+X13	(P 407 , P 941)
1 +X2 +X3 +X5 +X6 +X7 +X8 +X9 +X13	(P 409 , P 503)
1 +X2 +X3 +X5 +X8 +X10+X13	(P 411 , P 377)
1 +X2 +X6 +X7 +X8 +X9 +X10+X12+X13	(P 413 , P 71)
1 +X4 +X5 +X6 +X8 +X9 +X10+X12+X13	(P 415 , P 587)
1 +X4 +X7 +X8 +X9 +X10+X11+X12+X13	(P 419 , P1331)
1 +X2 +X8 +X9 +X13	(P 421 , P 349)
1 +X1 +X2 +X3 +X6 +X12+X13	(P 423 , P1461)

1 +X4 +X9 +X10+X13	(P 425 , P 763)
1 +X2 +X4 +X6 +X8 +X9 +X13	(P 427 , P 883)
1 +X2 +X4 +X5 +X8 +X10+X11+X12+X13	(P 429 , P 207)
1 +X1 +X3 +X9 +X10+X12+X13	(P 431 , P1239)
1 +X3 +X4 +X6 +X8 +X9 +X10+X11+X13	(P 433 , P 893)
1 +X2 +X3 +X4 +X5 +X7 +X8 +X11+X13	(P 435 , P1851)
1 +X1 +X3 +X4 +X6 +X9 +X10+X12+X13	(P 437 , P 479)
1 +X4 +X5 +X7 +X11+X12+X13	(P 439 , P1981)
1 +X2 +X8 +X9 +X10+X12+X13	(P 441 , P1023)
1 +X2 +X3 +X5 +X7 +X8 +X11+X12+X13	(P 443 , P 253)
1 +X1 +X2 +X3 +X9 +X10+X11+X12+X13	(P 445 , P 17)
1 +X1 +X4 +X10+X13	(P 447 , P 163)
1 +X1 +X3 +X4 +X5 +X6 +X10+X12+X13	(P 453 , P 275)
1 +X2 +X3 +X5 +X8 +X9 +X10+X12+X13	(P 455 , P 679)
1 +X2 +X4 +X5 +X6 +X9 +X11+X12+X13	(P 457 , P 101)
1 +X2 +X6 +X7 +X8 +X10+X13	(P 459 , P 937)
1 +X1 +X4 +X6 +X9 +X10+X11+X12+X13	(P 461 , P 337)
1 +X3 +X6 +X9 +X10+X11+X13	(P 463 , P1195)
1 +X2 +X3 +X5 +X8 +X9 +X11+X12+X13	(P 465 , P 331)
1 +X1 +X2 +X3 +X5 +X8 +X9 +X12+X13	(P 467 , P1453)
1 +X2 +X3 +X5 +X9 +X10+X13	(P 469 , P 739)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X12+X13	(P 471 , P1711)
1 +X1 +X3 +X5 +X7 +X8 +X9 +X10+X11+X12+X13	(P 473 , P 115)
1 +X2 +X4 +X6 +X7 +X11+X13	(P 475 , P 635)
1 +X3 +X5 +X7 +X9 +X12+X13	(P 477 , P 201)

1 +X1 +X2 +X6 +X7 +X8 +X9 +X12+X13	(P 479 , P 717)
1 +X2 +X3 +X4 +X7 +X10+X13	(P 483 , P1749)
1 +X1 +X3 +X4 +X10+X12+X13	(P 485 , P 869)
1 +X1 +X2 +X4 +X6 +X7 +X10+X11+X13	(P 487 , P2743)
1 +X1 +X2 +X6 +X7 +X8 +X13	(P 489 , P 359)
1 +X3 +X5 +X6 +X9 +X10+X13	(P 491 , P1399)
1 +X1 +X2 +X4 +X5 +X6 +X8 +X11+X13	(P 493 , P 467)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X10+X13	(P 495 , P1499)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X9 +X13	(P 497 , P 847)
1 +X1 +X3 +X4 +X9 +X10+X11+X12+X13	(P 499 , P2927)
1 +X1 +X2 +X3 +X4 +X7 +X8 +X11+X13	(P 501 , P 999)
1 +X1 +X2 +X3 +X8 +X9 +X10+X12+X13	(P 503 , P3063)
1 +X2 +X3 +X4 +X6 +X10+X11+X12+X13	(P 505 , P 61)
1 +X2 +X5 +X7 +X10+X12+X13	(P 507 , P 191)
1 +X4 +X7 +X9 +X10+X12+X13	(P 509 , P 133)
1 +X4 +X6 +X8 +X9 +X10+X11+X12+X13	(P 511 , P 391)
1 +X1 +X4 +X5 +X6 +X9 +X11+X12+X13	(P 547 , P1879)
1 +X1 +X3 +X4 +X8 +X9 +X10+X12+X13	(P 549 , P1389)
1 +X1 +X5 +X7 +X9 +X10+X11+X12+X13	(P 551 , P2911)
1 +X2 +X3 +X6 +X8 +X9 +X10+X11+X13	(P 553 , P 875)
1 +X1 +X2 +X3 +X4 +X6 +X10+X12+X13	(P 555 , P1915)
1 +X1 +X5 +X6 +X7 +X8 +X10+X11+X13	(P 557 , P 727)
1 +X1 +X4 +X6 +X8 +X10+X13	(P 559 , P1759)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X12+X13	(P 563 , P3823)
1 +X1 +X2 +X4 +X7 +X8 +X13	(P 565 , P1519)
1 +X1 +X2 +X3 +X5 +X9 +X11+X12+X13	(P 567 , P3583)

1 +X2 +X3 +X6 +X7 +X11+X13	(P 569 , P 251)
1 +X1 +X2 +X4 +X6 +X7 +X8 +X11+X13	(P 571 , P 511)
1 +X3 +X4 +X5 +X7 +X8 +X9 +X12+X13	(P 573 , P 99)
1 +X4 +X6 +X8 +X9 +X12+X13	(P 575 , P 57)
1 +X3 +X4 +X6 +X7 +X12+X13	(P 581 , P 615)
1 +X1 +X3 +X4 +X5 +X7 +X11+X12+X13	(P 583 , P 93)
1 +X1 +X2 +X7 +X9 +X10+X13	(P 585 , P 873)
1 +X1 +X3 +X6 +X9 +X12+X13	(P 587 , P 501)
1 +X1 +X4 +X10+X11+X12+X13	(P 589 , P 857)
1 +X2 +X6 +X7 +X9 +X11+X13	(P 591 , P 315)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X9 +X10+X11+X13	(P 595 , P 759)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X8 +X13	(P 597 , P1647)
1 +X2 +X3 +X4 +X7 +X10+X11+X12+X13	(P 599 , P 111)
1 +X2 +X3 +X4 +X5 +X6 +X8 +X9 +X13	(P 601 , P 631)
1 +X2 +X3 +X7 +X10+X11+X13	(P 603 , P1017)
1 +X3 +X7 +X9 +X13	(P 605 , P 461)
1 +X2 +X5 +X6 +X13	(P 607 , P 573)
1 +X3 +X6 +X7 +X13	(P 611 , P1275)
1 +X1 +X11+X12+X13	(P 613 , P1909)
1 +X1 +X2 +X4 +X11+X12+X13	(P 615 , P 351)
1 +X2 +X4 +X5 +X13	(P 617 , P1391)
1 +X3 +X5 +X10+X13	(P 619 , P1533)
1 +X1 +X2 +X4 +X6 +X10+X13	(P 621 , P 987)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X8 +X10+X12+X13	(P 623 , P 831)
1 +X4 +X7 +X8 +X9 +X10+X13	(P 627 , P1791)

1 +X2 +X4 +X5 +X8 +X11+X13	(P 629 , P2039)
1 +X2 +X5 +X6 +X7 +X9 +X11+X12+X13	(P 631 , P 15)
1 +X1 +X2 +X5 +X6 +X8 +X10+X11+X13	(P 633 , P 95)
1 +X2 +X8 +X12+X13	(P 635 , P 5)
1 +X2 +X3 +X4 +X5 +X7 +X9 +X10+X11+X12+X13	(P 637 , P 225)
1 +X3 +X4 +X6 +X9 +X10+X13	(P 639 , P 145)
1 +X1 +X2 +X3 +X5 +X8 +X10+X11+X13	(P 651 , P 85)
1 +X7 +X9 +X12+X13	(P 653 , P 745)
1 +X1 +X5 +X6 +X7 +X8 +X9 +X10+X11+X12+X13	(P 655 , P 405)
1 +X2 +X3 +X4 +X11+X12+X13	(P 659 , P 235)
1 +X1 +X2 +X4 +X7 +X9 +X10+X11+X13	(P 661 , P1005)
1 +X4 +X5 +X6 +X8 +X10+X13	(P 663 , P 369)
1 +X2 +X3 +X7 +X8 +X9 +X13	(P 665 , P 889)
1 +X1 +X3 +X10+X13	(P 667 , P 75)
1 +X2 +X4 +X5 +X6 +X7 +X9 +X10+X13	(P 669 , P 591)
1 +X2 +X3 +X4 +X5 +X9 +X11+X12+X13	(P 671 , P 665)
1 +X1 +X3 +X7 +X10+X11+X13	(P 675 , P 365)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X10+X11+X13	(P 677 , P1525)
1 +X2 +X3 +X4 +X7 +X8 +X10+X12+X13	(P 679 , P 795)
1 +X3 +X4 +X7 +X8 +X9 +X10+X12+X13	(P 681 , P1655)
1 +X1 +X5 +X6 +X7 +X8 +X10+X12+X13	(P 683 , P 215)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X8 +X9 +X12+X13	(P 685 , P1247)
1 +X3 +X4 +X8 +X10+X11+X13	(P 687 , P 925)
1 +X1 +X3 +X8 +X11+X12+X13	(P 691 , P 495)
1 +X1 +X2 +X3 +X4 +X10+X13	(P 693 , P2045)
1 +X3 +X5 +X7 +X8 +X12+X13	(P 695 , P 249)

1 +X2 +X4 +X5 +X6 +X8 +X10+X11+X13	(P 697 , P 509)
1 +X2 +X5 +X8 +X9 +X11+X13	(P 699 , P 35)
1 +X2 +X5 +X9 +X13	(P 701 , P 41)
1 +X3 +X4 +X8 +X9 +X10+X11+X12+X13	(P 703 , P 293)
1 +X5 +X6 +X10+X13	(P 711 , P 809)
1 +X3 +X4 +X5 +X8 +X10+X13	(P 713 , P 469)
1 +X3 +X4 +X5 +X7 +X9 +X11+X12+X13	(P 715 , P 345)
1 +X2 +X5 +X8 +X9 +X10+X11+X12+X13	(P 717 , P 299)
1 +X1 +X4 +X5 +X8 +X10+X11+X12+X13	(P 719 , P1325)
1 +X1 +X2 +X3 +X6 +X7 +X8 +X10+X13	(P 723 , P 755)
1 +X2 +X6 +X8 +X10+X11+X13	(P 725 , P 107)
1 +X2 +X8 +X11+X13	(P 727 , P 627)
1 +X2 +X3 +X4 +X5 +X6 +X8 +X11+X13	(P 729 , P 985)
1 +X1 +X2 +X6 +X7 +X8 +X9 +X10+X13	(P 731 , P 205)
1 +X1 +X3 +X4 +X5 +X6 +X9 +X11+X13	(P 733 , P 557)
1 +X2 +X7 +X8 +X9 +X11+X13	(P 735 , P1237)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X9 +X13	(P 739 , P 885)
1 +X4 +X5 +X6 +X9 +X11+X13	(P 741 , P 343)
1 +X1 +X2 +X5 +X6 +X11+X13	(P 743 , P1383)
1 +X2 +X4 +X7 +X8 +X9 +X10+X12+X13	(P 745 , P1501)
1 +X3 +X4 +X6 +X7 +X8 +X9 +X11+X13	(P 747 , P 475)
1 +X4 +X6 +X10+X13	(P 749 , P 815)
1 +X1 +X2 +X4 +X5 +X8 +X10+X11+X13	(P 751 , P1965)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X11+X13	(P 755 , P1015)
1 +X1 +X4 +X8 +X9 +X10+X13	(P 757 , P 59)

1 +X2 +X6 +X11+X13	(P 759 , P 189)
1 +X5 +X9 +X11+X13	(P 761 , P 319)
1 +X1 +X2 +X3 +X6 +X10+X11+X12+X13	(P 763 , P 135)
1 +X1 +X3 +X5 +X9 +X10+X11+X12+X13	(P 765 , P 393)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X11+X13	(P 767 , P 651)
1 +X1 +X2 +X6 +X7 +X10+X11+X12+X13	(P 793 , P1405)
1 +X1 +X2 +X3 +X8 +X11+X13	(P 795 , P 335)
1 +X2 +X7 +X8 +X11+X12+X13	(P 797 , P 851)
1 +X1 +X3 +X4 +X5 +X6 +X8 +X12+X13	(P 799 , P1367)
1 +X1 +X2 +X3 +X4 +X7 +X13	(P 805 , P 859)
1 +X3 +X4 +X5 +X6 +X7 +X8 +X10+X11+X12+X13	(P 807 , P1899)
1 +X1 +X2 +X4 +X6 +X7 +X9 +X10+X11+X12+X13	(P 809 , P2939)
1 +X2 +X3 +X4 +X6 +X7 +X8 +X9 +X10+X11+X13	(P 811 , P 735)
1 +X1 +X4 +X7 +X9 +X12+X13	(P 813 , P1767)
1 +X1 +X2 +X3 +X4 +X7 +X9 +X12+X13	(P 815 , P2799)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X10+X11+X12+X13	(P 819 , P1535)
1 +X4 +X6 +X9 +X10+X12+X13	(P 821 , P 247)
1 +X3 +X5 +X6 +X7 +X8 +X10+X12+X13	(P 823 , P 507)
1 +X1 +X4 +X6 +X8 +X9 +X10+X12+X13	(P 825 , P 767)
1 +X3 +X4 +X5 +X6 +X9 +X13	(P 827 , P 25)
1 +X1 +X2 +X4 +X5 +X8 +X9 +X11+X13	(P 829 , P 229)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X10+X13	(P 831 , P 179)
1 +X2 +X3 +X8 +X10+X11+X13	(P 839 , P 437)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X8 +X9 +X10+X13	(P 841 , P1003)
1 +X2 +X4 +X8 +X9 +X11+X13	(P 843 , P 283)

1 +X1 +X7 +X8 +X11+X12+X13	(P 845 , P1261)
1 +X3 +X5 +X6 +X7 +X11+X13	(P 847 , P 695)
1 +X1 +X2 +X5 +X9 +X12+X13	(P 851 , P 103)
1 +X2 +X6 +X7 +X9 +X10+X11+X12+X13	(P 853 , P 623)
1 +X2 +X3 +X6 +X7 +X9 +X13	(P 855 , P 953)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X11+X12+X13	(P 857 , P1663)
1 +X2 +X4 +X6 +X7 +X8 +X10+X12+X13	(P 859 , P 465)
1 +X2 +X4 +X7 +X8 +X9 +X13	(P 861 , P 981)
1 +X1 +X3 +X4 +X5 +X8 +X13	(P 863 , P1211)
1 +X3 +X4 +X5 +X6 +X8 +X13	(P 869 , P1375)
1 +X6 +X7 +X12+X13	(P 871 , P1469)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X8 +X11+X12+X13	(P 873 , P3067)
1 +X1 +X2 +X4 +X5 +X9 +X13	(P 875 , P 799)
1 +X3 +X5 +X6 +X9 +X11+X13	(P 877 , P2027)
1 +X1 +X2 +X4 +X7 +X8 +X9 +X11+X13	(P 879 , P1727)
1 +X1 +X3 +X4 +X5 +X7 +X9 +X12+X13	(P 883 , P 29)
1 +X2 +X3 +X5 +X7 +X8 +X9 +X11+X13	(P 885 , P 47)
1 +X2 +X6 +X7 +X10+X11+X13	(P 887 , P 159)
1 +X1 +X3 +X4 +X5 +X11+X13	(P 889 , P 7)
1 +X2 +X4 +X5 +X6 +X8 +X9 +X12+X13	(P 891 , P 265)
1 +X1 +X2 +X3 +X4 +X5 +X9 +X10+X11+X12+X13	(P 893 , P 177)
1 +X2 +X4 +X6 +X7 +X9 +X11+X12+X13	(P 895 , P 419)
1 +X1 +X3 +X4 +X5 +X8 +X10+X11+X13	(P 911 , P 939)
1 +X1 +X6 +X7 +X13	(P 915 , P 361)
1 +X4 +X6 +X7 +X10+X11+X13	(P 917 , P 567)

1 +X2 +X3 +X5 +X10+X11+X13	(P 919 , P1199)
1 +X1 +X3 +X5 +X6 +X7 +X8 +X10+X13	(P 921 , P 79)
1 +X1 +X2 +X3 +X5 +X6 +X7 +X8 +X9 +X10+X13	(P 923 , P 595)
1 +X4 +X6 +X7 +X8 +X9 +X11+X12+X13	(P 925 , P 697)
1 +X3 +X6 +X7 +X11+X12+X13	(P 927 , P1459)
1 +X2 +X5 +X10+X13	(P 933 , P 827)
1 +X3 +X7 +X8 +X13	(P 935 , P1719)
1 +X1 +X5 +X6 +X8 +X10+X13	(P 937 , P 223)
1 +X1 +X3 +X5 +X6 +X9 +X10+X11+X13	(P 939 , P1255)
1 +X3 +X5 +X7 +X8 +X9 +X13	(P 941 , P 957)
1 +X1 +X3 +X7 +X8 +X9 +X10+X12+X13	(P 943 , P1979)
1 +X1 +X2 +X9 +X11+X12+X13	(P 947 , P 245)
1 +X1 +X2 +X3 +X6 +X9 +X10+X11+X13	(P 949 , P 505)
1 +X1 +X2 +X6 +X8 +X9 +X10+X12+X13	(P 951 , P 765)
1 +X1 +X2 +X4 +X8 +X10+X11+X12+X13	(P 953 , P 9)
1 +X1 +X2 +X3 +X4 +X6 +X8 +X12+X13	(P 955 , P 165)
1 +X1 +X2 +X5 +X7 +X8 +X11+X12+X13	(P 957 , P 147)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X8 +X13	(P 959 , P 423)
1 +X1 +X2 +X6 +X8 +X9 +X13	(P 969 , P 267)
1 +X1 +X7 +X10+X11+X12+X13	(P 971 , P1197)
1 +X3 +X5 +X7 +X8 +X9 +X11+X12+X13	(P 973 , P 663)
1 +X1 +X4 +X6 +X8 +X9 +X11+X12+X13	(P 975 , P1455)
1 +X4 +X5 +X8 +X10+X12+X13	(P 979 , P 619)
1 +X6 +X7 +X8 +X13	(P 981 , P 921)
1 +X2 +X5 +X6 +X8 +X9 +X11+X12+X13	(P 983 , P1659)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X9 +X13	(P 985 , P 209)

1 +X2 +X4 +X6 +X8 +X11+X13	(P 987 , P 725)
1 +X3 +X4 +X6 +X8 +X9 +X11+X12+X13	(P 989 , P1179)
1 +X5 +X6 +X8 +X9 +X10+X11+X12+X13	(P 991 , P1757)
1 +X2 +X3 +X4 +X5 +X9 +X13	(P 997 , P1437)
1 +X5 +X7 +X12+X13	(P 999 , P3003)
1 +X3 +X6 +X12+X13	(P1001 , P 483)
1 +X1 +X3 +X5 +X6 +X7 +X8 +X10+X11+X12+X13	(P1003 , P1515)
1 +X1 +X2 +X3 +X5 +X12+X13	(P1005 , P1695)
1 +X1 +X2 +X5 +X6 +X8 +X13	(P1007 , P3519)
1 +X1 +X2 +X5 +X6 +X9 +X11+X12+X13	(P1011 , P 187)
1 +X1 +X5 +X6 +X7 +X10+X13	(P1013 , P 317)
1 +X2 +X4 +X10+X11+X12+X13	(P1015 , P 447)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X9 +X10+X11+X13	(P1017 , P 137)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X12+X13	(P1019 , P 395)
1 +X1 +X2 +X8 +X11+X12+X13	(P1021 , P 653)
1 +X3 +X4 +X5 +X7 +X10+X11+X12+X13	(P1023 , P 911)
1 +X1 +X3 +X6 +X8 +X9 +X10+X11+X13	(P1171 , P 599)
1 +X4 +X6 +X7 +X8 +X9 +X10+X12+X13	(P1173 , P1263)
1 +X2 +X4 +X7 +X8 +X10+X13	(P1175 , P 83)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X11+X12+X13	(P1179 , P 729)
1 +X3 +X4 +X5 +X6 +X7 +X11+X12+X13	(P1181 , P1523)
1 +X1 +X3 +X4 +X5 +X7 +X13	(P1183 , P 397)
1 +X3 +X4 +X5 +X6 +X8 +X9 +X10+X13	(P1189 , P1783)
1 +X4 +X8 +X9 +X11+X12+X13	(P1191 , P 231)
1 +X3 +X8 +X9 +X10+X11+X13	(P1195 , P 989)

1 +X1 +X2 +X3 +X5 +X8 +X13	(P1197 , P2043)
1 +X1 +X3 +X4 +X7 +X9 +X10+X12+X13	(P1199 , P 241)
1 +X1 +X2 +X3 +X5 +X6 +X8 +X9 +X13	(P1203 , P 761)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X12+X13	(P1205 , P1021)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X10+X11+X12+X13	(P1207 , P 37)
1 +X3 +X7 +X9 +X11+X12+X13	(P1211 , P 295)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X9 +X11+X12+X13	(P1213 , P 53)
1 +X2 +X4 +X6 +X7 +X8 +X9 +X12+X13	(P1215 , P 553)
1 +X1 +X2 +X5 +X6 +X7 +X10+X11+X13	(P1227 , P1327)
1 +X1 +X2 +X3 +X4 +X5 +X8 +X9 +X10+X11+X13	(P1229 , P 91)
1 +X2 +X5 +X6 +X8 +X9 +X10+X11+X13	(P1231 , P 611)
1 +X1 +X5 +X7 +X11+X12+X13	(P1235 , P1639)
1 +X2 +X3 +X4 +X7 +X9 +X10+X11+X13	(P1237 , P 493)
1 +X5 +X7 +X8 +X9 +X10+X13	(P1239 , P 213)
1 +X1 +X2 +X5 +X6 +X7 +X8 +X9 +X10+X11+X13	(P1243 , P1245)
1 +X2 +X5 +X7 +X8 +X9 +X10+X11+X13	(P1245 , P 311)
1 +X6 +X7 +X8 +X9 +X12+X13	(P1247 , P 917)
1 +X1 +X3 +X5 +X6 +X7 +X9 +X11+X13	(P1253 , P 751)
1 +X2 +X5 +X10+X11+X12+X13	(P1255 , P 491)
1 +X3 +X10+X12+X13	(P1259 , P2029)
1 +X2 +X3 +X6 +X10+X11+X13	(P1261 , P 55)
1 +X2 +X3 +X4 +X5 +X7 +X9 +X10+X13	(P1263 , P 185)
1 +X3 +X5 +X8 +X10+X11+X13	(P1267 , P 445)
1 +X1 +X4 +X6 +X7 +X9 +X10+X12+X13	(P1269 , P 575)
1 +X1 +X2 +X5 +X7 +X8 +X10+X11+X13	(P1271 , P 139)

1 +X3 +X4 +X7 +X8 +X9 +X11+X12+X13	(P1275 , P 655)
1 +X2 +X5 +X6 +X8 +X10+X11+X12+X13	(P1277 , P 569)
1 +X1 +X4 +X5 +X9 +X11+X13	(P1279 , P1171)
1 +X2 +X3 +X4 +X5 +X6 +X9 +X11+X13	(P1323 , P2815)
1 +X3 +X4 +X7 +X9 +X12+X13	(P1325 , P 239)
1 +X1 +X2 +X3 +X5 +X8 +X9 +X11+X13	(P1327 , P 499)
1 +X3 +X4 +X6 +X9 +X12+X13	(P1331 , P1019)
1 +X1 +X6 +X8 +X9 +X11+X13	(P1333 , P1279)
1 +X1 +X2 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X11+X12+X13	(P1335 , P 51)
1 +X3 +X8 +X10+X13	(P1339 , P 45)
1 +X1 +X2 +X3 +X5 +X6 +X8 +X9 +X11+X12+X13	(P1341 , P 489)
1 +X4 +X5 +X6 +X11+X12+X13	(P1343 , P 309)
1 +X2 +X5 +X9 +X10+X12+X13	(P1355 , P 87)
1 +X1 +X3 +X5 +X7 +X11+X13	(P1357 , P 607)
1 +X4 +X5 +X7 +X9 +X11+X13	(P1359 , P 825)
1 +X3 +X5 +X6 +X7 +X9 +X11+X12+X13	(P1363 , P 477)
1 +X1 +X3 +X5 +X10+X11+X13	(P1365 , P2037)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X11+X13	(P1367 , P 473)
1 +X2 +X3 +X4 +X6 +X9 +X10+X11+X13	(P1371 , P 303)
1 +X2 +X3 +X5 +X7 +X11+X13	(P1373 , P1343)
1 +X4 +X5 +X8 +X11+X12+X13	(P1375 , P1341)
1 +X1 +X4 +X8 +X11+X12+X13	(P1383 , P1011)
1 +X2 +X4 +X5 +X7 +X12+X13	(P1387 , P 27)
1 +X1 +X7 +X8 +X9 +X10+X11+X12+X13	(P1389 , P 23)
1 +X3 +X4 +X9 +X13	(P1391 , P 157)
1 +X1 +X3 +X4 +X7 +X9 +X13	(P1395 , P 287)

1 +X3 +X9 +X10+X11+X12+X13	(P1397 , P 11)
1 +X5 +X6 +X9 +X11+X12+X13	(P1399 , P 269)
1 +X2 +X3 +X6 +X9 +X11+X13	(P1403 , P 547)
1 +X2 +X4 +X5 +X7 +X8 +X9 +X10+X11+X12+X13	(P1405 , P 153)
1 +X1 +X3 +X5 +X7 +X10+X11+X12+X13	(P1407 , P 677)
1 +X1 +X2 +X5 +X6 +X12+X13	(P1431 , P 603)
1 +X1 +X2 +X6 +X8 +X12+X13	(P1435 , P 819)
1 +X3 +X5 +X6 +X7 +X10+X11+X12+X13	(P1437 , P 413)
1 +X1 +X3 +X5 +X6 +X7 +X10+X11+X13	(P1439 , P1717)
1 +X2 +X4 +X5 +X9 +X11+X13	(P1447 , P1271)
1 +X2 +X6 +X8 +X10+X12+X13	(P1451 , P 237)
1 +X1 +X4 +X6 +X9 +X10+X13	(P1453 , P 497)
1 +X4 +X5 +X11+X13	(P1455 , P 757)
1 +X2 +X3 +X4 +X5 +X8 +X9 +X10+X11+X12+X13	(P1459 , P1277)
1 +X5 +X8 +X10+X11+X12+X13	(P1461 , P 19)
1 +X3 +X4 +X5 +X6 +X7 +X9 +X10+X11+X12+X13	(P1463 , P 167)
1 +X8 +X9 +X11+X13	(P1467 , P 425)
1 +X1 +X2 +X3 +X10+X11+X13	(P1469 , P 277)
1 +X2 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X11+X13	(P1471 , P 683)
1 +X1 +X4 +X6 +X9 +X12+X13	(P1485 , P 793)
1 +X1 +X2 +X3 +X4 +X5 +X6 +X9 +X13	(P1487 , P1643)
1 +X1 +X2 +X3 +X4 +X5 +X8 +X11+X13	(P1491 , P1973)
1 +X1 +X4 +X5 +X6 +X7 +X8 +X9 +X11+X12+X13	(P1493 , P 217)
1 +X1 +X4 +X6 +X13	(P1495 , P 733)
1 +X1 +X2 +X3 +X9 +X12+X13	(P1499 , P1335)
1 +X2 +X3 +X5 +X6 +X7 +X8 +X12+X13	(P1501 , P 933)

1 +X1 +X2 +X5 +X13	(P1503 , P2747)
1 +X1 +X5 +X8 +X9 +X11+X13	(P1511 , P1531)
1 +X1 +X5 +X6 +X8 +X9 +X11+X12+X13	(P1515 , P 183)
1 +X3 +X5 +X6 +X10+X12+X13	(P1517 , P 313)
1 +X1 +X4 +X5 +X6 +X8 +X10+X12+X13	(P1519 , P 443)
1 +X1 +X6 +X7 +X8 +X10+X11+X12+X13	(P1523 , P 703)
1 +X1 +X9 +X11+X13	(P1525 , P 141)
1 +X3 +X5 +X6 +X8 +X9 +X10+X11+X13	(P1527 , P 399)
1 +X2 +X3 +X4 +X6 +X8 +X9 +X12+X13	(P1531 , P 915)
1 +X2 +X3 +X5 +X6 +X8 +X10+X12+X13	(P1533 , P1173)
1 +X1 +X7 +X8 +X9 +X10+X13	(P1535 , P1431)
1 +X1 +X2 +X3 +X6 +X7 +X8 +X9 +X10+X12+X13	(P1639 , P 13)
1 +X1 +X4 +X7 +X9 +X10+X13	(P1643 , P 39)
1 +X4 +X5 +X7 +X8 +X10+X11+X12+X13	(P1645 , P 221)
1 +X4 +X5 +X6 +X7 +X8 +X10+X11+X13	(P1647 , P 143)
1 +X1 +X2 +X5 +X10+X12+X13	(P1653 , P 271)
1 +X1 +X6 +X8 +X9 +X10+X11+X12+X13	(P1655 , P 273)
1 +X1 +X2 +X5 +X7 +X10+X11+X12+X13	(P1659 , P 169)
1 +X2 +X3 +X4 +X6 +X9 +X13	(P1661 , P 741)
1 +X1 +X5 +X10+X11+X12+X13	(P1663 , P 403)
1 +X2 +X3 +X6 +X7 +X8 +X11+X12+X13	(P1691 , P 429)
1 +X1 +X2 +X4 +X6 +X8 +X9 +X10+X13	(P1693 , P1781)
1 +X2 +X4 +X5 +X7 +X8 +X10+X11+X13	(P1695 , P 923)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X9 +X10+X11+X13	(P1703 , P 233)
1 +X1 +X2 +X4 +X6 +X7 +X8 +X10+X13	(P1707 , P 559)

1 +X1 +X3 +X4 +X8 +X11+X13	(P1709 , P1013)
1 +X2 +X3 +X7 +X13	(P1711 , P1183)
1 +X2 +X4 +X7 +X8 +X9 +X10+X11+X13	(P1717 , P 21)
1 +X1 +X2 +X3 +X4 +X6 +X7 +X9 +X11+X12+X13	(P1719 , P 297)
1 +X1 +X2 +X12+X13	(P1723 , P 555)
1 +X2 +X3 +X6 +X8 +X12+X13	(P1725 , P 171)
1 +X3 +X5 +X7 +X9 +X11+X13	(P1727 , P 813)
1 +X1 +X3 +X6 +X11+X12+X13	(P1743 , P1703)
1 +X2 +X3 +X4 +X5 +X10+X13	(P1749 , P 279)
1 +X6 +X9 +X10+X11+X12+X13	(P1751 , P1253)
1 +X2 +X3 +X4 +X5 +X7 +X10+X12+X13	(P1755 , P 949)
1 +X3 +X5 +X8 +X9 +X10+X13	(P1757 , P 687)
1 +X5 +X6 +X7 +X8 +X9 +X13	(P1759 , P1963)
1 +X1 +X2 +X3 +X6 +X7 +X11+X12+X13	(P1767 , P 181)
1 +X4 +X5 +X6 +X10+X11+X13	(P1771 , P 441)
1 +X4 +X5 +X7 +X8 +X9 +X10+X11+X13	(P1773 , P 571)
1 +X1 +X3 +X6 +X9 +X10+X13	(P1775 , P 701)
1 +X1 +X2 +X3 +X4 +X5 +X7 +X9 +X11+X12+X13	(P1781 , P 401)
1 +X1 +X2 +X11+X13	(P1783 , P 659)
1 +X1 +X2 +X4 +X5 +X7 +X8 +X9 +X11+X12+X13	(P1787 , P1175)
1 +X1 +X3 +X5 +X6 +X8 +X13	(P1789 , P1203)
1 +X1 +X3 +X5 +X7 +X9 +X10+X11+X13	(P1791 , P1691)
1 +X6 +X7 +X10+X13	(P1851 , P 155)
1 +X3 +X4 +X6 +X7 +X8 +X10+X12+X13	(P1853 , P 749)
1 +X1 +X4 +X6 +X7 +X12+X13	(P1855 , P 439)

1 +X1 +X5 +X6 +X7 +X11+X13	(P1871 , P 955)
1 +X1 +X3 +X5 +X6 +X7 +X8 +X12+X13	(P1877 , P 997)
1 +X2 +X3 +X6 +X10+X12+X13	(P1879 , P1213)
1 +X1 +X2 +X9 +X10+X11+X13	(P1883 , P 671)
1 +X2 +X3 +X5 +X8 +X10+X11+X12+X13	(P1885 , P2751)
1 +X6 +X8 +X11+X13	(P1887 , P1471)
1 +X2 +X3 +X4 +X6 +X7 +X8 +X9 +X10+X12+X13	(P1899 , P 285)
1 +X1 +X4 +X7 +X10+X12+X13	(P1901 , P 175)
1 +X2 +X3 +X4 +X7 +X9 +X10+X12+X13	(P1903 , P 415)
1 +X1 +X3 +X7 +X9 +X10+X11+X12+X13	(P1909 , P 305)
1 +X1 +X2 +X4 +X7 +X8 +X9 +X12+X13	(P1911 , P 675)
1 +X1 +X3 +X6 +X7 +X8 +X9 +X10+X13	(P1915 , P 805)
1 +X1 +X4 +X6 +X7 +X9 +X11+X12+X13	(P1917 , P 435)
1 +X3 +X5 +X8 +X13	(P1919 , P 935)
1 +X2 +X5 +X6 +X7 +X8 +X9 +X12+X13	(P1951 , P1975)
1 +X6 +X8 +X9 +X10+X11+X13	(P1963 , P1269)
1 +X1 +X3 +X5 +X8 +X9 +X11+X12+X13	(P1965 , P1215)
1 +X7 +X8 +X10+X11+X12+X13	(P1967 , P1789)
1 +X1 +X3 +X4 +X5 +X8 +X9 +X10+X13	(P1973 , P 149)
1 +X2 +X3 +X5 +X6 +X10+X13	(P1975 , P 427)
1 +X3 +X5 +X6 +X8 +X10+X11+X12+X13	(P1979 , P 685)
1 +X2 +X3 +X6 +X7 +X8 +X13	(P1981 , P 407)
1 +X2 +X4 +X5 +X6 +X9 +X10+X12+X13	(P1983 , P 943)
1 +X2 +X9 +X10+X13	(P2005 , P1181)
1 +X2 +X5 +X6 +X7 +X12+X13	(P2007 , P1773)

1 +X1 +X2 +X3 +X5 +X7 +X13	(P2011 , P2735)
1 +X1 +X2 +X3 +X6 +X7 +X8 +X9 +X10+X11+X13	(P2013 , P1439)
1 +X1 +X2 +X7 +X8 +X10+X13	(P2015 , P3007)
1 +X1 +X4 +X5 +X6 +X7 +X8 +X12+X13	(P2027 , P 699)
1 +X4 +X7 +X12+X13	(P2029 , P 829)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X9 +X11+X12+X13	(P2031 , P 959)
1 +X1 +X2 +X3 +X5 +X7 +X8 +X9 +X10+X11+X13	(P2037 , P 661)
1 +X1 +X2 +X4 +X5 +X6 +X8 +X9 +X10+X12+X13	(P2039 , P 919)
1 +X1 +X2 +X4 +X8 +X10+X13	(P2043 , P1435)
1 +X4 +X5 +X6 +X8 +X12+X13	(P2045 , P1693)
1 +X3 +X4 +X6 +X8 +X9 +X13	(P2047 , P1951)
1 +X2 +X3 +X8 +X10+X12+X13	(P2731 , P 43)
1 +X1 +X2 +X3 +X5 +X6 +X9 +X11+X13	(P2735 , P 301)
1 +X1 +X3 +X4 +X5 +X6 +X9 +X10+X13	(P2743 , P 563)
1 +X5 +X6 +X7 +X9 +X10+X11+X12+X13	(P2747 , P 409)
1 +X2 +X9 +X10+X11+X12+X13	(P2751 , P1333)
1 +X2 +X3 +X5 +X6 +X7 +X8 +X9 +X10+X11+X13	(P2775 , P 173)
1 +X1 +X3 +X4 +X5 +X6 +X7 +X9 +X13	(P2779 , P 433)
1 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X13	(P2783 , P 693)
1 +X2 +X3 +X4 +X6 +X9 +X11+X12+X13	(P2795 , P 151)
1 +X2 +X3 +X5 +X6 +X10+X11+X12+X13	(P2799 , P 667)
1 +X2 +X4 +X9 +X10+X11+X13	(P2807 , P 821)
1 +X1 +X3 +X4 +X8 +X10+X11+X12+X13	(P2811 , P 669)
1 +X1 +X2 +X5 +X6 +X7 +X9 +X12+X13	(P2815 , P2731)
1 +X3 +X4 +X5 +X7 +X8 +X9 +X10+X13	(P2907 , P 281)

1 +X1 +X3 +X4 +X7 +X8 +X9 +X12+X13	(P2911 , P 411)
1 +X1 +X3 +X4 +X6 +X7 +X9 +X11+X13	(P2927 , P 797)
1 +X2 +X3 +X4 +X5 +X7 +X10+X11+X13	(P2935 , P1191)
1 +X1 +X4 +X5 +X11+X12+X13	(P2939 , P1189)
1 +X2 +X4 +X5 +X6 +X7 +X8 +X11+X13	(P2943 , P1451)
1 +X1 +X2 +X3 +X4 +X7 +X8 +X10+X13	(P2991 , P 431)
1 +X1 +X6 +X7 +X9 +X10+X13	(P2999 , P 947)
1 +X1 +X3 +X4 +X5 +X9 +X10+X12+X13	(P3003 , P1205)
1 +X1 +X3 +X6 +X7 +X10+X13	(P3007 , P1463)
1 +X1 +X3 +X5 +X6 +X8 +X9 +X10+X13	(P3035 , P 691)
1 +X1 +X3 +X6 +X7 +X9 +X10+X12+X13	(P3039 , P 951)
1 +X2 +X3 +X4 +X5 +X11+X13	(P3055 , P 927)
1 +X2 +X3 +X4 +X6 +X8 +X9 +X10+X13	(P3063 , P1853)
1 +X5 +X6 +X9 +X10+X11+X13	(P3067 , P1709)
1 +X2 +X5 +X6 +X9 +X12+X13	(P3071 , P2991)
1 +X3 +X4 +X6 +X11+X12+X13	(P3511 , P1207)
1 +X2 +X3 +X8 +X13	(P3519 , P1723)
1 +X1 +X2 +X4 +X7 +X9 +X13	(P3551 , P1467)
1 +X1 +X3 +X4 +X6 +X7 +X8 +X10+X11+X12+X13	(P3567 , P1447)
1 +X1 +X3 +X5 +X8 +X11+X13	(P3575 , P1725)
1 +X1 +X5 +X6 +X7 +X8 +X13	(P3583 , P3511)
1 +X3 +X4 +X12+X13	(P3823 , P1707)
1 +X2 +X3 +X4 +X5 +X6 +X7 +X8 +X9 +X10+X11+X12+X13	(P3839 , P2999)
1 +X2 +X3 +X8 +X9 +X10+X11+X12+X13	(P3967 , P1967)
1 +X1 +X3 +X4 +X6 +X7 +X8 +X12+X13	(P4031 , P1983)
1 +X9 +X10+X12+X13	(P4095 , P4031)

Appendix B

COMPUTER PROGRAM FOR POLYNOMIALS LISTS OF APPENDIX A

```
C
C
C       THIS PROGRAM GENERATES THE CYCLOTOMIC COSETS MODULO N,
C   WHERE N = 2**M-1, THEN ACCORDING TO THE APPROPRIATE CHOICE
C   OF THE 'LANDA' VALUE DEPICTED IN THE SECTION 3.1 OF THE THESIS,
C   TOGETHER WITH VARIOUS LOGIC ASSIGNMENTS,
C   WE SEARCH OUT THOSE OPTIMAL POLYNOMIAL PAIRS
C   AND LIST OUT THEM. TO DO THIS, WE READ IN THE
C   OCTAL IRREDUCIBLE POLYNOMIAL DATA OF
C    ' PETERSON & WELDON : ERROR-CORRECTING CODES, 2ND EDITION ',
C   AND TRANSFORM THEM INTO BINARY AND POLYNOMIAL FORMS.
C
C
ISN 0002            INTEGER COSET(1200,14),KOSET(600,6),LEDRK(600),BELTA
ISN 0003            INTEGER ARRY0(3,8),ARRY1(15),ARRY2(15),ARRY3(15),BLANK
ISN 0004            LOGICAL COND1,COND2,COND3
C
ISN 0005            DATA BLANK/'     '/
ISN 0006            DATA ARRY0/0,0,1,0,1,0,0,1,1,1,0,0,
                   +           1,0,1,1,1,0,1,1,1,0,0,0/
ISN 0007            DATA ARRY1/'+X14','+X13','+X12','+X11','+X10',
                   +           '+X9','+X8','+X7','+X6','+X5',
                   +           '+X4','+X3','+X2','+X1','  1'/
C
C
ISN 0008            DO 280 M=3,14
ISN 0009            I=0
ISN 0010            N=(2**M)-1
ISN 0011            COSET(1,1)=1
C
C       DEFINE THE LANDA VALUE FOR EACH DEGREE M.
C
ISN 0012            COND1=(M.EQ.3).OR.(M.EQ.5).OR.(M.EQ.7).OR.
                   +      (M.EQ.9).OR.(M.EQ.11).OR.(M.EQ.13)
ISN 0013            COND2=(M.EQ.6).OR.(M.EQ.10).OR.(M.EQ.14)
ISN 0014            COND3=(M.EQ.4).OR.(M.EQ.8).OR.(M.EQ.12)
ISN 0015            IF(COND1) LANDA=2**((M+1)/2)+1
ISN 0017            IF(COND2) LANDA=2**((M+2)/2)+1
ISN 0019            IF(COND3) LANDA=2**((M+2)/2)-1
C
C       THE FIRST PART IS USED FOR GENERATING THE CYCLOTOMIC COSETS,
C
ISN 0021          1 I=I+1
ISN 0022          2 DO 3 J=2,M
ISN 0023            L=2*COSET(I,J-1)
ISN 0024            COSET(I,J)=MOD(L,N)
ISN 0025          3 IF(COSET(I,J).LT.COSET(I,1)) GO TO 4
ISN 0027            COSET(I+1,1)=COSET(I,1)+2
ISN 0028            IF(COSET(I+1,1).GT.(N+1)/2) GO TO 5
ISN 0030            GO TO 1
ISN 0031          4 COSET(I,1)=COSET(I,1)+2
ISN 0032            GO TO 2
```

```
C
C       PRINT OUT THE ' DEGREE ' TITLE AT THE TOP OF A NEW PAGE.
C
ISN 0033      5 WRITE(6,6) M
ISN 0034      6 FORMAT('1',47X,'DEGREE',I3/1X,'------------------------------',
             +   '------------------------------------------------------')
C
C       READ IN THE DATA CARDS NUMBER FOR THE SPECIFIED DEGREE M.
C
C
ISN 0035        READ(5,7) NUM,INDX
ISN 0036      7 FORMAT(I5,70X,I5)
ISN 0037        IF(INDX.NE.99999) STOP
C
C       READ IN THE IRREDUCIBLE POLYNOMIAL DATA,
C   SET THE COSET LEADER IN 'LEDRK'.
C
ISN 0039        DO 9 I1=1,NUM
ISN 0040        READ(5,8) (KOSET(I1,J1),J1=1,6)
ISN 0041      8 FORMAT(I5,2X,5I1)
ISN 0042      9 LEDRK(I1)=KOSET(I1,1)
C
C
C       THE SECOND PART FIRSTLY CHECK THE SPECIFIED COSET
C   WHETHER SUITABLE FOR ASSOCIATING AN M-SEQUENCE CODE,
C   THEN ACCORDING TO VARIOUS LOGIC, WE SEARCH OUT
C   THOSE PREFERRED POLYNOMIAL PAIRS.
C       ALSO, BASED ON THE KNOWN IRREDUCIBLE POLYNOMIAL DATA
C   WE TRANSFORM THEM INTO BINARY FORMS AND THEN FURTHER INTO
C   POLYNOMIAL FORMS.
C
C
ISN 0043        DO 260 I2=1,I
ISN 0044        KT=COSET(I2,1)
C
C       FOR EVERY DEGREE M, CHECK WHETHER THE COSET LEADER
C   RELATIVELY PRIME TO N, IF NOT, THEN FORGET THAT COSET.
C
ISN 0045        K=M-2
ISN 0046        GO TO (18,10,18,11,18,12,13,14,15,16,18,17),K
C
ISN 0047      10 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0049        IF(MOD(KT,5).EQ.0) GO TO 260
ISN 0051        GO TO 18
C
ISN 0052      11 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0054        IF(MOD(KT,7).EQ.0) GO TO 260
ISN 0056        GO TO 18
C
ISN 0057      12 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0059        IF(MOD(KT,5).EQ.0) GO TO 260
ISN 0061        IF(MOD(KT,17).EQ.0) GO TO 260
ISN 0063        GO TO 18
```

```
                C
ISN 0064           13 IF(MOD(KT,7).EQ.0) GO TO 260
ISN 0066              IF(MOD(KT,73).EQ.0) GO TO 260
ISN 0068              GO TO 18
                C
ISN 0069           14 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0071              IF(MOD(KT,11).EQ.0) GO TO 260
ISN 0073              IF(MOD(KT,31).EQ.0) GO TO 260
ISN 0075              GO TO 18
                C
ISN 0076           15 IF(MOD(KT,23).EQ.0) GO TO 260
ISN 0078              IF(MOD(KT,89).EQ.0) GO TO 260
ISN 0080              GO TO 18
                C
ISN 0081           16 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0083              IF(MOD(KT,5).EQ.0) GO TO 260
ISN 0085              IF(MOD(KT,7).EQ.0) GO TO 260
ISN 0087              IF(MOD(KT,13).EQ.0) GO TO 260
                      GO TO 18
ISN 0089        C
                C
ISN 0090           17 IF(MOD(KT,3).EQ.0) GO TO 260
ISN 0092              IF(MOD(KT,43).EQ.0) GO TO 260
ISN 0094              IF(MOD(KT,127).EQ.0) GO TO 260
                C
                C     ACCORDING TO THE LOGIC OF '' BELTA**1 = ALPHA**KT,
                C  BELTA**LANDA = ALPHA**(KT*LANDA MOD N) '',
                C  WE SEARCH OUT THE PREFERRED POLYNOMIAL PAIRS
                C  L1 AND L2 WHICH ARE MINIMAL POLYNOMIALS HAVING
                C  BELTA AND BELTA**LANDA AS RESPECTIVE ROOTS.
                C
ISN 0096           18 L1=KT
ISN 0097              BELTA=MOD(KT*LANDA,N)
                C
ISN 0098              DO 20 I1=1,I
ISN 0099              IF(I1.EQ.L1) GO TO 20
ISN 0101              DO 20 J1=1,M
ISN 0102              IF(COSET(I1,J1).EQ.BELTA) GO TO 30
ISN 0104           20 CONTINUE
ISN 0105           30 L2=COSET(I1,1)
                C
                C     PRE-CLEAR THE MEMORY ARRY3 WHICH IS USED FOR THE STORAGE OF THE
                C  IRREDUCIBLE POLYNOMIALS BEFORE THEIR PRINTED OUT.
                C
ISN 0106              DO 40 K=1,15
ISN 0107           40 ARRY3(K)=BLANK
```

```
C
C
C     THE FOLLOWING PART CONVERTS THE READ-IN DATA INTO BINARY FORMS
C   AND THEN FURTHER INTO POLYNOMIAL FORMS. IT DIVIDES INTO 2 SECTIONS:
C      1)  THE ESTABLISHED COSET LEADERS CAN BE MATCHED WITH
C   THOSE OF THE READ-IN  (STATEMENT NUMBER 200 TO 240),
C   SO THAT WE HAVE A REGULAR POLYNOMIAL TRACING.
C      2)  THE OTHER SECTION IS FOR THE UNMATCHED CASE AND
C   CORRESPONDS TO THE RECIPROCAL POLYNOMIAL TRACING
C   ( STATEMENT NUMBER 100 TO 190 ).
C
C
ISN 0108          IF(KT.GT.LEDRK(NUM)) GO TO 100
ISN 0110          DO 50 I5=1,NUM
ISN 0111       50 IF(KT.EQ.LEDRK(I5)) GO TO 200
C
C     FIND OUT THE MAXIMUM VALUE 'MAX1' IN THE SPECIFIED COSET,
C   THEN THIS COSET IS THE RECIPROCAL OF THAT COSET WITH
C   LEDRC = N - MAX1 AS LEADER, SO WE CAN TRACE OUT ITS DATA
C   AND PROCEED THE SUCCEEDING TRANSFORMATIONS.
C
ISN 0113       100 MAX1=KT
ISN 0114          DO 110 J2=2,M
ISN 0115       110 IF(COSET(I2,J2).GT.MAX1) MAX1=COSET(I2,J2)
ISN 0117          LEDRC=N-MAX1
ISN 0118          DO 120 I3=1,NUM
ISN 0119       120 IF(LEDRK(I3).EQ.LEDRC) GO TO 130
C
C     TRANSFORM THE READ-IN DATA INTO BINARY FORM.
C
ISN 0121       130 K1=0
ISN 0122          DO 150 J1=2,6
ISN 0123          K=KOSET(I5,J1)
ISN 0124          IF(K.EQ.0) K=6
ISN 0126          DO 140 I1=1,3

ISN 0127          K1=K1+1
ISN 0128       140 ARRY2(K1)=ARRY0(I1,K)
ISN 0129       150 CONTINUE
C
C     TRANSFORM THE BINARY DATA INTO THE IRREDUCIBLE POLYNOMIAL FORM,
C   THE DIRECTION IS FROM LEFT TO RIGHT
C   FOR THE RECIPROCAL POLYNOMIAL TRACING.
C
ISN 0130          DO 160 I1=1,15
ISN 0131          IF(ARRY2(I1).NE.0) GO TO 170
ISN 0133       160 ARRY2(I1)=BLANK
C
ISN 0134       170 DO 180 K1=I1,15
ISN 0135          IF(ARRY2(K1).EQ.0) ARRY2(K1)=BLANK
ISN 0137          IF(ARRY2(K1).EQ.1) ARRY2(K1)=ARRY1(15-K1+I1)
ISN 0139       180 CONTINUE
```

```
C
C     COMPRESS THE POLYNOMIAL ARRAY INTO CONTINUOUS OCCUPATION FORMAT.
C  THIS STEP CAN FACILITATE THE PRINT OUT OF THE POLYNOMIALS.
C
ISN 0140          K2=0
ISN 0141          DO 190 I1=1,15
ISN 0142          IF(ARRY2(I1).EQ.BLANK) GO TO 190
ISN 0144          K2=K2+1
ISN 0145          ARRY3(K2)=ARRY2(I1)
ISN 0146        190 CONTINUE
ISN 0147          GO TO 250
C
C     TRANSFORM THE READ-IN DATA INTO BINARY FORM.
C
ISN 0148        200 K1=0
ISN 0149          DO 220 J1=2,0
ISN 0150          K=KOSET(I3,J1)
ISN 0151          IF(K.EQ.0) K=6
ISN 0153          DO 210 I1=1,5
ISN 0154          K1=K1+1
ISN 0155        210 ARRY2(K1)=ARRY0(I1,K)
ISN 0156        220 CONTINUE
C
C     TRANSFORM THE BINARY DATA INTO THE IRREDUCIBLE POLYNOMIAL FORM,
C  THE DIRECTION IS FROM RIGHT TO LEFT
C  FOR THE REGULAR POLYNOMIAL TRACING.
C
ISN 0157          DO 230 K1=1,15
ISN 0158          IF(ARRY2(K1).EQ.0) ARRY2(K1)=BLANK
ISN 0160          IF(ARRY2(K1).EQ.1) ARRY2(K1)=ARRY1(K1)
ISN 0162        230 CONTINUE
C
C     COMPRESS THE POLYNOMIAL ARRAY INTO CONTINUOUS OCCUPATION FORMAT.
C  THIS STEP CAN FACILITATE THE PRINT OUT OF THE POLYNOMIALS.
C
ISN 0163          K2=0
ISN 0164          DO 240 I1=1,15
ISN 0165          K1=16-I1
ISN 0166          IF(ARRY2(K1).EQ.BLANK) GO TO 240
ISN 0168          K2=K2+1
ISN 0169          ARRY3(K2)=ARRY2(K1)
ISN 0170        240 CONTINUE
C
C      ACCORDING TO THE DIFFERENT M, CONTROL THEIR PRINT OUT FORMAT,
C  THEREFORE WE CAN GET A VERY PRETTY APPEARANCE.
C
C
ISN 0171        250 IF((M.EQ.7).OR.(M.EQ.8)) GO TO 252
ISN 0173          IF((M.EQ.9).OR.(M.EQ.10)) GO TO 254
ISN 0175          IF((M.EQ.11).OR.(M.EQ.12)) GO TO 250
ISN 0177          IF((M.EQ.13).OR.(M.EQ.14)) GO TO 250
```

```
                C
ISN 0179            WRITE(6,251) (ARRY3(K),K=1,7),L1,L2
ISN 0180          251 FORMAT('0',24X,7A4,'¦',7X,'( P',I2,' , P',I2,' )')
ISN 0181            GO TO 260
                C
ISN 0182          252 WRITE(6,253) (ARRY3(K),K=1,8),L1,L2
ISN 0183          253 FORMAT('0',20X,8A4,'¦',7X,'( P',I3,' , P',I3,' )')
ISN 0184            GO TO 260
                C
ISN 0185          254 WRITE(6,255) (ARRY3(K),K=1,10),L1,L2
ISN 0186          255 FORMAT('0',12X,10A4,'¦',7X,'( P',I3,' , P',I3,' )')
ISN 0187            GO TO 260
                C
ISN 0188          256 WRITE(6,257) (ARRY3(K),K=1,11),L1,L2
ISN 0189          257 FORMAT('0',8X,11A4,'¦',5X,'( P',I4,' , P',I4,' )')
ISN 0190            GO TO 260
                C
ISN 0191          258 WRITE(6,259) (ARRY3(K),K=1,13),L1,L2
ISN 0192          259 FORMAT('0',13A4,'¦',5X,'( P',I4,' , P',I4,' )')
                C
ISN 0193          260 CONTINUE
ISN 0194          280 CONTINUE
                C
                C
ISN 0195            STOP
ISN 0196            END
```

REFERENCES

1. Abramson, N., "Information Theory and Coding", New York: McGraw-Hill, 1963.

2. Baumert, L.D. and R.J. McEliece, "Weights of irreducible cyclic codes", Inform. Control, Vol. 20, pp. 158-175, March 1972.

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

4. Chakrabarti, N.B. and M. Tomlinson, "Design of sequences with specified autocorrelation and crosscorrelation", IEEE Trans. Commun., Vol. COM-24, pp. 1246-1252, November 1976.

5. Dixon, R.C., "Spread Spectrum Systems", New York: Wiley, 1976.

6. Fredricsson, S., "Pseudo-randomness properties of binary shift register sequences", IEEE Trans. Inform. Theory, Vol. IT-21, pp. 115-120, January 1975.

7. Gallager, R. G., "Information Theory and Reliable Communication", New York: Wiley, 1968.

8. Gold, R., "Optimal binary sequences for spread spectrum multiplexing", IEEE Trans. Inform. Theory, Vol. IT-13, pp. 619-621, October 1967.

9. Gold, R., "Maximal recursive sequences with 3-valued recursive cross-correlation functions", IEEE Trans. Inform. Theory, Vol. IT-14, pp. 154-156, January 1968.

10. Golomb, S.W., Ed., "Digital Communications with Space Applications", Englewood Cliffs, N.J.: Prentice-Hall, 1964.

11. Golomb, S.W., "Shift Register Sequences", San Francisco, CA.: Holden-Day, 1967.

12. Helleseth, T., "Some results about the cross-correlation function between two maximal linear sequences", Discrete Math., Vol. 16, pp. 209-232, November 1976.

13. Kasami, T., "Weight distribution of Bose-Chaudhuri-Hocquenghem codes", in Combinatorial Mathematics and Its Applications, R.C. Bose and T.A. Dowling, Eds., Chapel Hill, N.C.: Univ. of North Carolina Press, 1969. Reprinted in Key Papers in the Development of Coding Theory, E.R. Berlekamp, Ed., New York: IEEE Press, 1974.

14. Kotov, Y.I., "Correlation function of composite sequences constructed from two m-sequences", Radio Eng. Electron. Phys., Vol. 19, pp. 128-130, September 1974.

15. Lempel, A. and H. Greenberger, "Families of sequences with optimal Hamming correlation properties", IEEE Trans. Inform. Theory, Vol. IT-20, pp. 90-94, January 1974.

16. McEliece, R.J., "Correlation properties of sets of sequences derived from irreducible cyclic codes", Inform. Control, Vol. 45, pp. 18-25, April 1980.

17. MacWilliams, F.J. and N.J.A. Sloane, "Pseudo-random sequences and arrays", Proc. IEEE, Vol. 64, pp. 1715-1729, December 1976.

18. MacWilliams, F.J. and N.J.A. Sloane, "The Theory of Error-Correcting Codes", Amsterdam, the Netherlands: North-Holland, 1977.

19. Milstein, L.B., "Some statistical properties of combination sequences", IEEE Trans. Inform. Theory, Vol. IT-23, pp. 254-258, March 1977.

20. Milstein, L.B. and R.R. Ragonetti, "Combination sequences for spread spectrum communications", IEEE Trans. Commun., Vol. COM-25, pp. 691-696, July 1977.

21. Peterson, W.W. and E.J. Weldon, Jr., "Error-Correcting Codes, 2nd Edition." Cambridge, MA.: M.I.T. Press, 1972.

22. Pursley, M.B. and D.V. Sarwate, "Evaluation of correlation parameters for periodic sequences", IEEE Trans. Inform. Theory, Vol. IT-23, pp. 508-513, July 1977.

23. Pursley, M.B. and D.V. Sarwate, "Performance evaluation for phase-coded spread-spectrum multiple-access communication---Part I: System analysis, Part II: Code sequence analysis", IEEE Trans. Commun., Vol. COM-25, pp. 795-803, August 1977.

24. Sarwate, D.V., "Bounds on crosscorrelation and autocorrelation of sequences", IEEE Trans. Inform. Theory, Vol. IT-25, pp. 720-724, November 1979.

25. Sarwate, D.V. and M.B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences", Proc. IEEE, Vol. 68, pp. 593-619, May 1980.

26. Shannon, C.E., "A Mathematical Theory of Communication", Bell Syst. Tech. J., Vol. 27, pp. 379-423 and pp. 623-656, July and October 1948.

27. Shiva, S.G.S., P.E. Allard and G. Seguin, "A class of composite codes", IEEE Trans. Inform. Theory, Vol. IT-27, pp. 260-262, March 1981.

28. Titsworth, R.C., "Optimal ranging codes", IEEE Trans. Space Electron. Telem., Vol. SET-10, pp. 19-30, March 1964.

29. Turyn, R., "Sequences with small correlation", in Error Correcting Codes, H.B. Mann, Ed., New York: Wiley, 1968.

30. Wainberg, S. and J.K. Wolf, "Subsequences of pseudorandom sequences", IEEE Trans. Commun., Vol. COM-18, pp. 606-612, October 1970.

31. Welch, L.R., "Lower bounds on the maximum cross-correlation of signals", IEEE Trans. Inform. Theory, Vol. IT-20, pp. 397-399, May 1974.

32. Zierler, N., "Linear recurring sequences", J. Soc. Industrial and Applied Mathematics, Vol. 7, pp. 31-48, March 1959.

33. Zierler, N., "Linear recurring sequences and error-correcting codes", in Error Correcting Codes, H.B. Mann, Ed., New York: Wiley, 1968.