

THEORIES AND APPLICATIONS
OF PARALLEL LINEAR FEEDBACK
SHIFT REGISTER

By
MU-YUE HSIAO

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
December, 1967

~~XXXXXXXXXX~~-by

Mu-Yue Hsiao

1967

ACKNOWLEDGMENTS

The author wishes to express his deep appreciation to Dr. W. H. Chen and Dr. T. S. George for their guidance during the course of the research reported herein. He also wishes to thank Dr. D. G. Childers for pointing out some valuable references.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	iii
LIST OF FIGURES AND TABLES	vi
LIST OF SYMBOLOGY	ix
LIST OF TERMINOLOGY	x
ABSTRACT	xii
 CHAPTERS	
I. INTRODUCTION	1
1.1 Introduction to the subject	1
1.2 Remarks on the linear and nonlinear feedback shift register	6
1.3 Review of pertinent prior work on LFSR	7
1.4 Brief summary on the new result of this dissertation	8
II. MATHEMATICAL TRANSFORMATION TECHNIQUES	9
2.1 On serial to parallel transformation of a LFSR	9
2.2 The general state space approach	10
2.3 The parallel LFSR as a shift-register sequence generator (SRSR)	13
2.4 The parallel LFSR as a Galois field (GF) divider	19
2.5 The polynomial root approach to the transformation problem	23
III. PARALLEL LFSR AS A COMPLETE SEQUENTIAL MACHINE	25
3.1 Basic definitions	25
3.2 The equivalent machine and similar transformation ...	29
VI. PERIODS AND CHARACTERISTIC EQUATIONS OF THE PARALLEL LFSR.	36
4.1 On periods of the parallel LFSR	36
4.2 Methods of finding the characteristic equation of an autonomous f-channel analogy parallel LFSR	39
4.3 Concluding remarks	48

TABLE OF CONTENTS - Continued

	Page
V. PARALLEL LFSR IN A F-CHANNEL SYSTEM --IMPLEMENTATION OF CYCLIC ERROR CORRECTING CODES	51
5.1 The use of single error correcting cyclic code in a f-channel system	51
5.2 A straightforward 2-channel BCH (15,7) decoder	59
5.3 Burst error correction in a f-channel transmission system	64
VI. SINGLE CHANNEL ERROR CORRECTION IN A F-CHANNEL SYSTEM	68
6.1 General system description	68
6.2 The 2-dimensional variable length code (2-DVLC)	71
6.3 Mathematical justification	77
6.4 An example for illustration	78
6.5 Reverse transmission and reciprocal polynomial	85
6.6 Comments on Brown and Sellers' scheme	86
VII. PARALLEL GENERATION OF PN SEQUENCES	89
7.1 Generating PN sequences in parallel	89
7.2 A detail example	92
7.3 Basic theorems on the structure of the parallel PN sequences	95
7.4 Correlation and orthogonality of the block sequence..	100
PROPOSED FUTURE RESEARCH	102
APPENDIX 1.A	104
APPENDIX 1.B	107
REFERENCES	109
BIOGRAPHICAL SKETCH	113

LIST OF FIGURES AND TABLES

Figure	Page
1.1.a	1
1.1.b	1
1.2	1
1.3	2
1.4	3
1.5	5
2.1 A serial LFSR	9
2.2 A f-Channel Analogy Parallel LFSR	10
2.3	12
2.4 $g(x) = 1 + x^3 + x^4 + x^5 + x^6$	16
2.5 The 3-Channel Analogy SRSG for $g(x) = 1 + x^3 + x^4 + x^5 + x^6$	18
2.6 GF Divider $g(x) = 1 + x^3 + x^4 + x^5 + x^6$, $f=1$	21
2.7 GF Divider $g(x) = 1 + x^3 + x^4 + x^5 + x^6$, $f=3$	23
3.1 A GF Divider	26
3.2 State Diagram for $g(x) = 1 + x + x^2$, $f=2$	28
3.3	34
3.4	35
4.1 Serial LFSR $g(x) = 1 + x^2 + x^4$ and its Cycle Set	36
4.2 Cycle Set = $\{1(1), 5(3)\}$	37
5.1 Encoder and a Codeword	51

LIST OF FIGURES AND TABLES - Continued

Figure	Page
5.2 Decoder	53
5.3 2-Channel Analogy GF Divider $g(x) = 1 + x + x^4$	55
5.4 2-Channel Decoder for Hamming (15,11) Code	57
5.5 The 2-Channel Analogy SRSG $g(x) = x^8 + x^7 + x^6 + x^4 + 1$	60
5.6 3-Channel Analogy GF Divider $g(x) = 1 + x^4 + x^6 + x^7$	66
6.1	68
6.2 General System Diagram	69
6.3 Encoder	69
6.4 Decoder	70
6.5 A Codeword Block	71
6.6 Decoding Procedures in Flow-Diagram Form	73
6.7 $e(x)$ and $e'(x)$ in a Codeword Block	74
6.8 Decoder	80
6.9 4-Channel Analogy GF Divider $[g_1(x)]_1$ $g_1(x) = x^8 + x^5 + x^3 + x + 1$	81
6.10 Parallel Mode of $[g_1(x)]_2$ $g_1(x) = x^8 + x^5 + x^3 + x + 1$	82
6.11 Serial Mode of $[g_1(x)]_2$ $g_1(x) = x^8 + x^5 + x^3 + x + 1$	82
6.12	87
7.1 The Serial SRSG Characterized by $g(x)$	90
7.2 SRSG for $g(x) = 1 + x + x^4$	92
7.3 An Autonomous 4-Channel Analogy SRSG	94
7.4	98
7.5	99

LIST OF FIGURES AND TABLES - Continued

Table		Page
1.1	4
2.1	19
3.1	27
4.1	All Powers of α	44
4.2	All Powers of α for $g(x) = 1 + x^2 + x^4$	45
4.3	50
5.1	Error Correction State Table	58
5.2	Format of $(T^2)^{7-l} L E'_{jq}$	62
5.3	Double Error Pattern	63
5.4	Decoding Process Table	67
6.1	83
7.1	States Transition of the Serial SRSG	93
7.2	State Transition of the Circuit of Figure 7.3	94
7.3	95

SYMBOLOLOGY



: Storage device, or shift-register cell



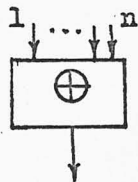
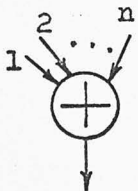
: The shift-register cell, which is corresponding to the x^i -th position of a polynomial $g(x)$, is indicated as shown.



: "AND" gate



: "OR" gate



: n -input modulo 2 adder, when $n=2$, it is the "EXCLUSIVE OR" gate.



: To specify the sequential circuit is at the state 4.



: "INVERTER" gate

TERMINOLOGY

A: similar transformation matrix

B, C, D, T, L, R, G, J, Q: The characterizing matrices of linear
feedback shift registers.

a_t^{i-1} : input symbol to the i -th channel at the time t .

b_t^{i-1} : output symbol of the i -th channel at the time t .

$E(x)$: error pattern polynomial.

E_{ti} : single error vector at the time t .

$e(x)$: error polynomial $= x^l E(x)$

$e'(x)$: shifted error polynomial $= x^{l-j} E(x)$

$F(x)$: codeword polynomial of degree $(n-1)$

f : number of parallel channels

$g(x)$: generator polynomial

$H(x)$: received polynomial $= F(x) + e(x)$

(n,k) code: codeword contains k information bits, total length is n .

$m(x)$: minimum polynomial

$m_i(x)$: the minimum polynomial that corresponds to the i -th power of
the root α .

P_s : one period of a serial LFSR

$GF(2)$: Galois field of 2 elements 0 and 1.

$GF(2^m)$: extension field of degree m over $GF(2)$, or Galois field of 2^m
elements.

$GF_2[x]$: the polynomial domain containing of all polynomial in x
with coefficients in $GF(2)$.

LFSR: linear feedback shift register

SRSRG: shift register sequence generator

GF divider: Galois field divider

P_p : the corresponding period (to P_s) of the parallel LFSR

$q(x)$: quotient polynomial

$r(x)$: remainder polynomial

r : the number of check bits, the degree of $g(x)$

S_t : $S_t = (x_t^0, x_t^1, \dots, x_t^{r-1})$ is the state vector of the LFSR at the time t .

T : the companion matrix of $g(x)$

U_t : $U_t = (a_t^0, a_t^1, \dots, a_t^{f-1})$ is the input vector of the parallel LFSR at the time t .

Y_t : $Y_t = (b_t^0, b_t^1, \dots, b_t^{f-1})$ is the output vector of the f -channel analogy circuit at time t . (Note: f -channel analogy circuit \equiv parallel LFSR)

Z : final state vector

W : general error pattern vector

α : a root of $g(x)$.

$\phi(\lambda)$: the characteristic polynomial of the matrix T

$\phi_f(\lambda)$: the characteristic polynomial of the matrix T^f

$\rho(t)$: autocorrelation function of a block sequence

ρ_{xy} : cross-correlation function

Φ : zero element

Φ : null matrix

$\zeta_i(T)$: the i -th row of the matrix T

$\eta_j(T)$: the j -th column of the matrix T

" , ": the symbol ' stands for transpose

Abstract of Dissertation Presented to the Graduate Council
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

THEORIES AND APPLICATIONS
OF PARALLEL LINEAR FEEDBACK SHIFT REGISTER

By

Mu-Yue Hsiao

December 1967

Chairman: Dr. W. H. Chen

Major Department: Electrical Engineering

This paper reports the results of an investigation of the theory and applications of binary parallel linear feedback shift registers (LFSR). Most existing results are on the conventional serial LFSR, i.e., the case of one input and one output. The parallel LFSR has f (f integer and $f > 0$) inputs and outputs. It performs the same operation as the serial LFSR but in a speed f -times faster. The amount of hardware to implement this parallel LFSR increases fractionally as compared with that of the serial LFSR. This parallel LFSR is precisely defined as the f -channel analogy circuit.

Result of investigation is divided into two parts, theory and applications. The theoretical part includes basic serial-to-parallel transformation techniques, which cover the Galois field divider and the shift register sequence generator. Next, the algebraic structure of the parallel LFSR as a complete linear sequential machine is shown. The important result in the theoretical part is related to the periods and the characteristic equation of a parallel LFSR. Three basic theorems are proven which convert easily all periods of a serial LFSR into the periods of its f -channel analogy circuit. The characteristic equation of a parallel LFSR is obtained by finding the equation $|T^f - I\lambda| = 0$. Two methods in addition to the direct computation method are discovered.

Results of the second part are applications. In the error correcting codes, we apply the parallel LFSR to implement various parallel encoding and decoding systems, such as single error correcting Hamming code, double error correcting BCH code, and burst error correcting Fire code. One special kind of error correcting system is also treated in detail which corrects unlimited errors in one channel in a multi-channel transmission system. In this case, variable length data blocks are encoded in terms of the entire block instead of the conventional case of encoding block data bits column-wise or row-wise. The resulting error correcting system is very flexible, economic and fast. It is very suitable in correcting errors in a tape, disk or drum system.

Next, the parallel LFSR can be used in deep space communication such as generating parallel PN sequences. The result of this investigation shows that a new method is obtained which not only generates PN sequences in blocks but also produces row PN sequences at the same time.

In summary, results of this investigation are readily applicable to data transmission, information processing and deep space communication systems.

CHAPTER 1

INTRODUCTION

1.1 Introduction to the subject

The problem investigated in this report is the theories and applications of the binary linear feedback shift register (LFSR) in parallel form. The system has multiple inputs and outputs in contrast with the conventional case of one input and one output. A r -stage serial LFSR specified by a polynomial $g(x)$,

$$g(x) = g_0 + g_1 x + g_2 x^2 + \dots + g_r x^r \quad (1.1)$$

is shown in Figures 1.1.a and 1.1.b and 1.2.

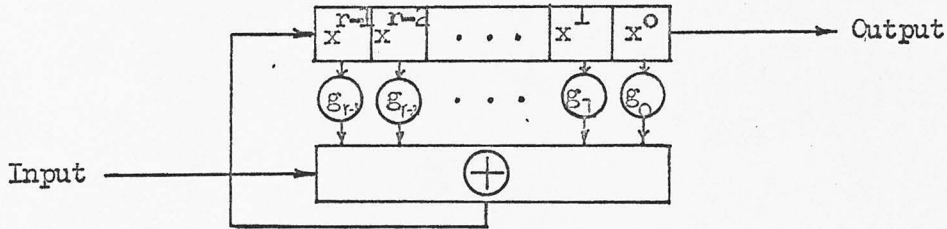


Figure 1.1.a

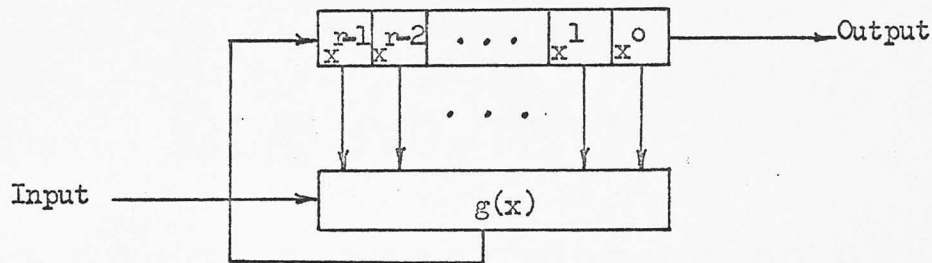


Figure 1.1.b

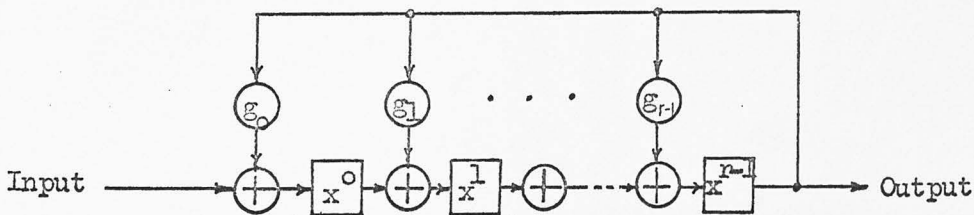


Figure 1.2

Each circuit contains r binary storage devices which imply that the maximum number of states is equal to 2^r . Each state S_i , $i=1,2,\dots,2^r-1$, is a r -tuple. In Figure 1.1, $S=(x^{r-1}, x^{r-2}, \dots, x^1, x^0)$. Since the circuit is specified by $g(x)$ with co-efficients $g_i \in GF(2)$ for all $i=0,1,\dots,r$, the output of the storage device x^i is tapped to the modulo 2 adder only if $g_i = 1$. Therefore, under the condition of input equals to zero, the state transition of S_i to S_j is completely specified by $g(x)$. The circuit of Figure 1.1 is generally named shift register sequence generator (SMSG) for input equal to zero, and that of Figure 1.2 is called the Galois field (GF) divider. Both circuits have the same amount of hardware and perform equivalent operations. However, these two circuits differ in certain places such as state transition and output sequences for the same $g(x)$ and input sequences. Moreover, the place of the mod 2 adder also makes a difference. From a practical point of view, the circuit of Figure 1.1 is better in a sense that the shift register can use the integrated circuit of several stages built in together.

The general form of the parallel LFSR specified by a polynomial $g(x)$ and the number of parallel channels is shown in Figure 1.3. Normally, the number f of the parallel channels is assumed to be less than or equal to the degree r of the polynomial $g(x)$.

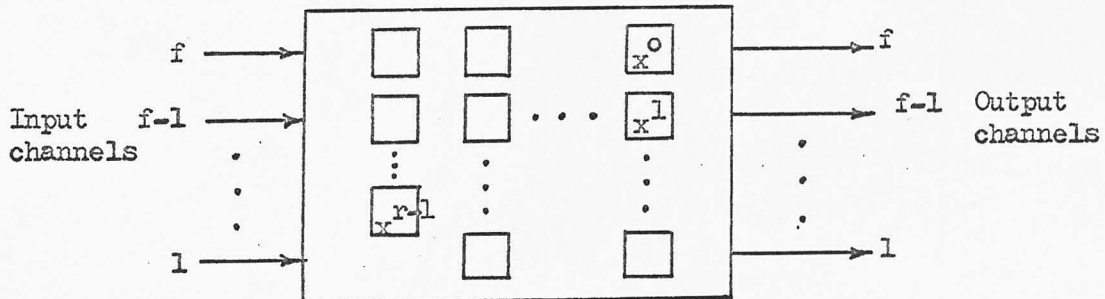


Figure 1.3

Figure 1.3 is the parallel form of Figure 1.1. Note the labeling of the storage device positions. However, interconnections among the mod 2 adders are no longer simply decided by $g(x)$. A full treatment is given in Chapter 2. The general circuit of Figure 1.4 is the parallel form of the circuit of Figure 1.2. The interconnections of the mod 2 adders are again different from that of Figure 1.3 for the same $g(x)$ and f .

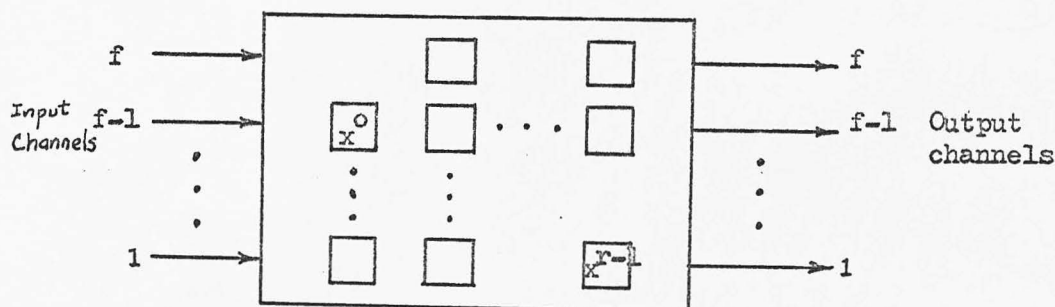


Figure 1.4

As a result of comparing the above figures, it is seen that the parallel LFSR is designed for a multi-channel information processing system to perform error correction and sequence generation operation. It is obviously better than the conventional serial LFSR in a parallel system because of the following reasons:

1. The operation speed performed by the parallel LFSR is f -times faster than the serial LFSR, where f is a positive integer.
2. The hardware of the parallel LFSR is increased fractionally as compared with that of the serial LFSR. However, the direct usage of the parallel LFSR eliminates the requirement of a parallel-to-serial buffer register which is normally needed in the case of using the serial LFSR. Therefore, the increasing importance of the parallel information processing system makes the parallel LFSR become a more important processing unit in the future.

In the following Table 1.1, various properties of the parallel LFSR which is transformed from a serial LFSR specified by $g(x)$ are compared with that of the original serial LFSR.

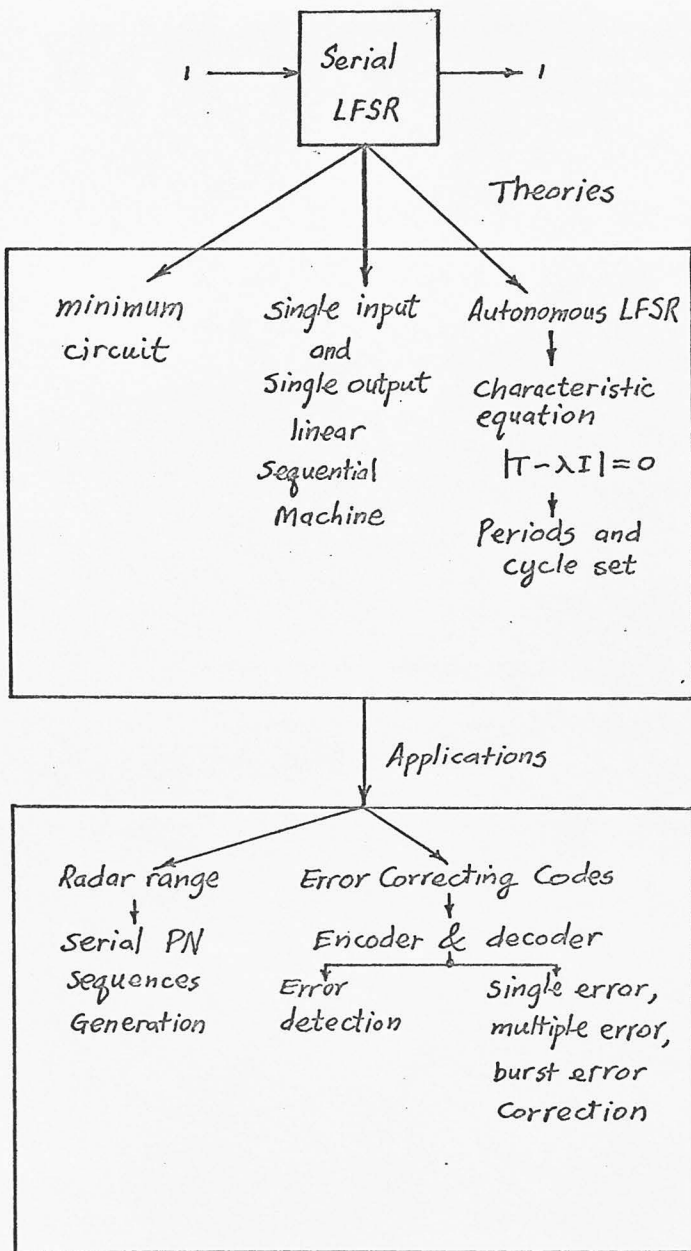
Table 1.1

A serial LFSR specified by $g(x)$	The parallel LFSR transformed from the given serial LFSR $g(x)$ (f-channel analogy circuit)
1. r-stage storage devices	1. r-stage storage devices
2. # of exclusive-or gate determined by $g(x)$	2. # of exclusive-or gate no longer specified by $g(x)$
3. Circuit connections specified by $g(x)$	3. Circuit connections no longer specified by $g(x)$
4. States transition as $S_1 \rightarrow S_2 \rightarrow S_3 \dots \rightarrow S_{2^r-1}$	4. States transition as $S_1 \rightarrow S_{f+1} \rightarrow S_{2f+1} \rightarrow \dots$
5. *For no-input condition, the circuit is determined by the matrix T	5. For no-input condition, the circuit is determined by the matrix T^f
6. As a GF divider, $v(x) = q(x) g(x) + r(x)$	6. Parallel GF divider, $v(x) = q(x) g(x) + r(x)$
7. As a SRSG, it produces required sequences	7. As a parallel SRSG, it produces required sequences in parallel provided certain conditions hold.

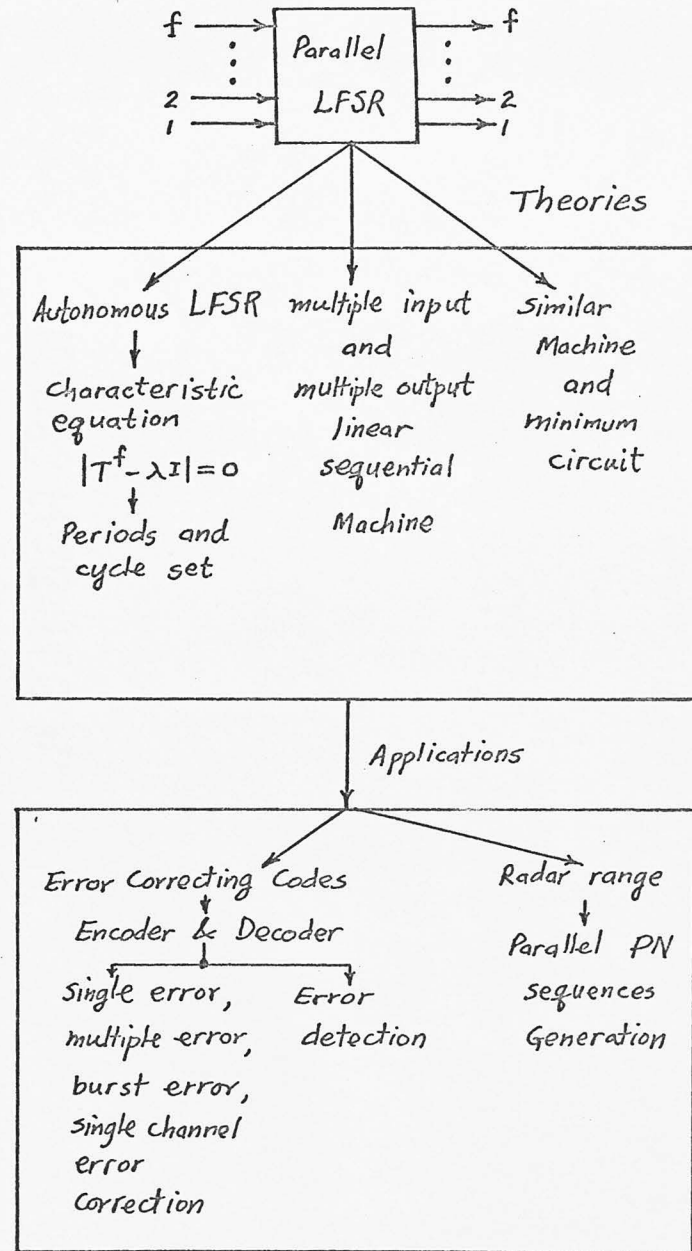
The f-channel parallel circuit, defined more precisely in Chapter 2, is the f-channel analogy circuit. Throughout this report, these two basic kinds of parallel LFSR, the SRSG and GF divider, are treated in detail. The block diagram of Figure 1.5 compares the existing result with the problem area to be investigated in this research. Namely,

1. Serial-to-parallel transformation techniques: Given a serial LFSR implemented as a SRSG or as a GF divider, find its f-channel analogy circuit.
2. Algebraic structure of the parallel LFSR as a linear sequential machine: The algebraic structure of this new kind of circuit is investigated on the basis of the existing knowledge of the linear sequential machines.

* The companion matrix T is discussed in Chapter 2.



existing results area



Research area

Figure 1.5

3. The period and the characteristic polynomial of the parallel LFSR as relating to the period and characteristic polynomial of the original serial LFSR.
4. Applications in the error correcting codes: This is to apply the parallel LFSR to implement various error correcting codes, such as single error correcting Hamming code, double error correcting BCH code, and burst error correcting code. One special kind of error correcting system is also treated in detail which corrects unlimited errors in one channel in a multi-channel transmission system.
5. Application to radar ranging: The existing methods of generating PN by using parallel LFSR.

1.2 Remarks on the linear and nonlinear feedback shift register

The basic circuits shown in Section 1.1 are linear circuits. A circuit is said to be linear over a finite field $GF(2)^*$ if its states form a vector space over $GF(2)$ and its next state and output are linear functions of the present state and input.^[1] An equivalent definition^[2] is that the states of the circuit can be identified with the elements of a vector space and its state and output behavior, described by a pair of matrix equations over $GF(2)$

$$S_{t+1} = S_t T + U_t C \quad (1.2.a)$$

$$Y_t = S_t B + U_t D \quad (1.2.b)$$

The vectors S_t , U_t and Y_t denote, respectively, the state, input, and output of the circuit $\{T, B, C, D\}$ at time t .

An alternative way of classifying linear and nonlinear circuits is to consider the basic logic elements involved in the circuits. Peterson^[3] uses three kinds of basic circuit devices. In $GF(2)$, only two kinds of devices are needed, i.e., the mod 2 adder and a storage device. The nonlinear circuit can be noted, e.g., the circuit in Figure 1.1.b where the polynomial $g(x)$ is no longer a linear com-

* $GF(2)$ can be replaced by $GF(p)$ for any positive prime integer p .

bination equation of the intermediate x^0, x^1, \dots, x^r . In general, if the logic operation involved in the $g(x)$ may contain AND or OR gates, it is a nonlinear feedback shift register. The existing result on serial nonlinear feedback shift registers may be referred to Magleby, [4] Yoeli, [5] Golomb et al. [6]

1.3 Review of pertinent prior work on LFSR

The theory of the serial LFSR has been studied and developed extensively. Early results on this subject are due to Zierler, [7] [8] Huffman, [9] Elspas, [10] and Birdsall and Ristenbatt [11] ... etc. The first four papers and many other good papers on this subject are now in a book edited by Kautz. [12] Other books relating to this subject are Peterson, [3] Golomb, [13] [14] Besides the above mentioned references, there exist numerous articles in relation to the theory and applications of the serial LFSR. These papers are listed at the end of the report. [15] → [21]

Because of the growing importance of the parallel transmission system in processing information both in the computer and communication systems, the serial LFSR no longer meets all requirements. The first hardware model of parallel LFSR was implemented on IBM system 360/2820 drum system in 1964. The result was due to simulation; no theory was established then. In fact, the first general theory paper was written by the author and Sih [22] in the same year. However, during the period of 1964 to 1967, only three papers have been published on this subject; two of them again due to Sih and the author, [23] [24] another one due to Gill. [25] The main reason may be that the advance of a completely parallel information process system still takes time to achieve.

Besides the above mentioned papers on the parallel LFSR, there exists a brief discussion on the multiple input and output linear switching system given by Peterson (p. 133-134 of [3]).

1.4 Brief summary on the new results of this dissertation

The main results are in Chapters 2, 3, 4, 5, 6, 7.

1. Chapter 2: This chapter presents the existing papers of the author and Sih and Gill in a unified form. It integrates the mathematical material in a systematic manner with examples for practical applications.
2. Chapter 3: Most material in this chapter is derived in this research. A new model of input and output symbols are given which applies the theory of finite state linear sequential machines. The section on similar transformation and equivalent machine also includes some practical implications.
3. Chapter 4: The result of this chapter is new and very useful. The basic problem of finding the period and the characteristic equation of a parallel LFSR is solved such that the result can be applied elsewhere. The new method given in this chapter in calculating all powers of a companion matrix T is convenient and can be applied in the coding theory.
4. Chapter 5: Part of the material in this chapter was previously published by the author and Sih. However, the presentation given here is more unified and systematic. The result on decoding double error correcting BCH (15,7) code is new.
5. Chapter 6: All material in this chapter is derived by the author in this research. The error correcting system utilizes the principles of parallel LFSR in a very interesting way. It generalizes the error correcting system used in the IBM 360/2400 tape system in which the basic principle is using the serial LFSR.
6. Chapter 7: All works in this chapter are new. The result of generating block PN sequences intermixing with row PN sequences is not only of theoretical interest, but also has a practical application. Theorems relating the generation of parallel PN sequences are derived.

CHAPTER 2

MATHEMATICAL TRANSFORMATION TECHNIQUES

2.1 On serial to parallel transformation of a LFSR

In Section 1.1, a general description of a serial LFSR and its f -channel analogy parallel LFSR is given. Mathematically, if a serial LFSR is connected as shown in Figure 1.1, it is any mapping T of binary r -space Ω_r into itself which satisfies the relation

$$T(a_1, a_2, \dots, a_r) = (a_2, a_3, \dots, a_{r+1})$$

Each vector of Ω_r is assigned a unique successor by the mapping T .^[13] Similarly, its f -channel analogy parallel LFSR is any mapping T^f of the same space into itself which satisfies the relation

$$T^f(a_1, a_2, \dots, a_r) = (a_{1+f}, a_{2+f}, \dots, a_{r+f})$$

Each vector of Ω_r is again a unique successor by the mapping T^f .

Definition 2.1: A serial LFSR K is a LFSR of equivalently, a linear sequential circuit, with one input and one output terminal.

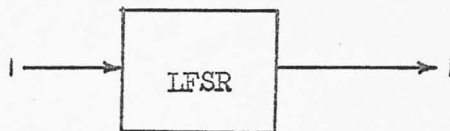


Figure 2.1 A Serial LFSR

Definition 2.2: A f -channel analogy of K or a f -channel analogy parallel LFSR, denoted by K^f , is a LFSR with f -input terminals and f -output terminals, whose input and output vectors correspond to input and output sequences of length f in K .

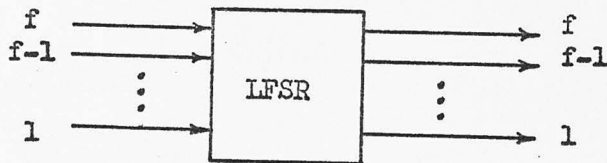


Figure 2.2 A f-Channel Analogy Parallel LFSR

More precisely, let the initial state of K be S_0 , its input a_t and output b_t at time t . Let the input of K^f at time t ($t = 0, 1, 2, \dots$) be

$$U'_t = \begin{bmatrix} a_{tf} \\ a_{tf+1} \\ \vdots \\ a_{tf+f-1} \end{bmatrix} = \begin{bmatrix} a_{tf}^0 \\ a_{tf}^1 \\ \vdots \\ a_{tf}^{f-1} \end{bmatrix}$$

where the symbol ' is to denote "transpose."

Then, with the same initial state S_0 , the output of K^f at time t is

$$Y'_t = \begin{bmatrix} b_{tf} \\ b_{tf+1} \\ \vdots \\ b_{tf+f-1} \end{bmatrix} = \begin{bmatrix} b_{tf}^0 \\ b_{tf}^1 \\ \vdots \\ b_{tf}^{f-1} \end{bmatrix}$$

Therefore, by employing f "channels" instead of one, K^f is capable of operating f times faster than K .

Section 2.2 discusses briefly the result of transformation of Gill^[31] who generalizes the early result of Hsiao and Sih^[22] to consider a LFSR as a Galois field divider. Section 2.3 presents a general version of considering a LFSR as a shift-register sequence generator. Section 2.4 treats the Galois field divider differently.

2.2 The general state space approach

The r -dimensional vector space Ω_r over Galois field characterized 2, denoted by $GF(2)$, is used to describe the operation of a r -stage LFSR

since all 2^r possible states can be defined as vectors of Ω_r and satisfy all properties of a vector space. The Ω_r is called the state space; and S_t , the state vector at time t ,

$$S_t = (x_t^0, x_t^1, \dots, x_t^{r-1})$$

The subscript t in the r -tuple can be omitted if the case is obvious.

Similarly, a f -dimensional input/output space Ω_f over $GF(2)$ can be also defined. The input vector U_t and the output vector Y_t specified at time t are as follows:

$$U_t = (a_t^0, a_t^1, \dots, a_t^{f-1})$$

$$Y_t = (b_t^0, b_t^1, \dots, b_t^{f-1})$$

Again, the subscript t can be omitted for an obvious situation.

The following set of equations completely specifies the characteristics of a serial LFSR.

$$S'_{t+1} = T S'_t + B U'_t \quad (2.1.a)$$

$$Y'_t = C S'_t + D U'_t \quad (2.1.b)$$

where T is a $r \times r$, B is a $r \times f$, C is a $f \times r$ and D is a $f \times f$ matrix. In the following sections, $f = 1$ is the case used in Equations (2.1.a) and (2.1.b). They are all defined over $GF(2)$.

Definition 2.3: A serial LFSR is called autonomous if the input vector $U_t = 0$, i.e., under no input condition.

Substituting $U_t = 0$ in Equations (2.1.a) and (2.1.b), we have

$$S'_{t+1} = T S'_t \quad (2.2.a)$$

$$Y'_t = C S'_t \quad (2.2.b)$$

Also $S'_{t+j} = T^j S'_t \quad (2.2.c)$

Equation (2.2.a) implies that the state transition of an autonomous serial LFSR is completely determined by the characteristic matrix T . T is also called as the next state operator since $TS'_t = S'_{t+1}$. The following equivalent relations are due to Golomb. [13]

- (i) Every state vector S_t has a predecessor S_{t-1} as well as a successor S_{t+1} .
- (ii) Predecessors of vectors, when they exist, are unique.
- (iii) Distinct vectors have distinct successors.

In general, the matrix T is specified by a polynomial $g(x)$,

$$g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \dots + g_r x^r$$

and usually bears the names of companion matrix, [26] associated matrix, [27] and characteristic matrix [28] of $g(x)$. It can be one of the following four forms.

$$\begin{bmatrix} 0 & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & I_{r-1} & & \\ g_0 & g_1 & \dots & g_{r-1} & \end{bmatrix}_{r \times r}$$

(a)

$$\begin{bmatrix} 0 & 0 & \dots & 0 & g_0 \\ & & & & g_1 \\ & & & & \vdots \\ I_{r-1} & & & & g_{r-1} \end{bmatrix}_{r \times r}$$

(b)

$$\begin{bmatrix} g_{r-1} & g_{r-2} & \dots & g_0 & \\ & & & 0 & \\ & & & 0 & \\ & & I_{r-1} & \vdots & \\ & & & 0 & \end{bmatrix}_{r \times r}$$

(c)

$$\begin{bmatrix} g_{r-1} & & & & \\ g_{r-2} & & & & \\ \vdots & & I_{r-1} & & \\ g_0 & 0 & 0 & \dots & 0 \end{bmatrix}_{r \times r}$$

(d)

Figure 2.3

where I_{r-1} is the identity matrix of order $(r-1)$.

Actually, these four matrices are similar to each other. In the following discussions, only the companion matrix form of Figure 2.3.a,

is fully discussed. Only this form is used to specify the two general classes of LFSR, the shift-register sequences generator and the Galois field divider. The difference lies in how the connections of the circuit is interrupted from the companion matrix. The general serial-to-parallel transformation theory discovered by Gill^[31] generalizes our early result^[22] and is independent of the form of the T matrix.

Theorem 2.1: Let K be a serial LFSR with the characterizing matrices T, B, C and D, then an f-channel analogy of K is the parallel LFSR K^f with characterizing equations:

$$S'_{t+1} = T^f S'_t + L U'_t \quad (2.3.a)$$

$$Y'_t = R S'_t + Q U'_t \quad (2.3.b)$$

and T^f is the f-th power of T,

$$L = [T^{f-1} B, T^{f-2} B, \dots, TB, B]$$

$$R = \begin{bmatrix} C \\ CT \\ CT^2 \\ \vdots \\ CT^{f-1} \end{bmatrix}, \quad Q = \begin{bmatrix} D & & & \\ CB & D & & \\ CTB & CB & D & \\ \vdots & & \ddots & \\ CT^{f-2} B & CT^{f-3} B & \dots & CB D \end{bmatrix} \quad (2.4)$$

Proof: See Gill.^[31]

This theorem applies well to the following two sections.

2.3 The parallel LFSR as a shift-register sequence generator (SRSG)

The name shift-register sequence generator, or shift-register generator, is adapted from Zierler,^[7] Peterson,^[3] and Golomb.^[13] It is used here merely to indicate the case that all modulo-2 adders are out of the storage devices as shown in Figure 1.1. In the next section, the modulo-2 adders are placed in between storage devices. The case used there is the Galois field divider. In general, the SRSG and the Galois field divider are equivalent in many respects, such as they both can be used to

generate sequences or encoding and decoding cyclic codes. Results on using serial SRSg to implement cyclic codes have been given by Melas,^[29] Meggitt,^[27] and Abramson,^[30]. A more general description is in Ash.^[28]

In this section, the problem is for a given serial SRSg, to find its f -channel analogy SRSg. Through the result may be obtained directly from Equations (2.2.a) and (2.2.b), all individual matrix must be carefully specified. In the serial case, using the form of Figure 2.3.a, we have

$$g(x) = g_0 + g_1 x + g_2 x^2 + \dots + g_r x^r \Rightarrow T = \begin{bmatrix} x_t^0 & x_t^1 & \dots & x_t^{r-1} \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \\ g_0 & g_1 & \dots & g_{r-1} \end{bmatrix} \begin{bmatrix} x_{t+1}^0 \\ x_{t+1}^1 \\ \vdots \\ x_{t+1}^{r-2} \\ x_{t+1}^{r-1} \end{bmatrix} \quad (2.5)$$

The connection of the SRSg is read out row-wise and is rewritten as follows:

$$\begin{bmatrix} x_{t+1}^0 \\ x_{t+1}^1 \\ \vdots \\ x_{t+1}^{r-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & \vdots & & \ddots \\ & g_0 & g_1 & \dots & g_{r-1} \end{bmatrix} \begin{bmatrix} x_t^0 \\ x_t^1 \\ \vdots \\ x_t^{r-1} \end{bmatrix} \quad (2.6.a)$$

$$\text{or} \quad S'_{t+1} = T S'_t \quad (2.6.b)$$

Next, as reading directly from Figure 1.1, we have

$$B = \begin{bmatrix} x^0 \\ x^1 \\ \vdots \\ x^{r-1} \end{bmatrix}_{rx1}, \quad C = \begin{bmatrix} x^0 & x^1 & \dots & x^{r-1} \\ 1 & 0 & \dots & 0 \end{bmatrix}_{l \times r}, \quad D = [0]_{1 \times 1} \quad (2.7)$$

Letting $\gamma_i(T)$ and $\eta_j(T)$ denote the i -th row and j -th column of the matrix T , respectively, we have

$$L = [\eta_r^{a^0}(T^{f-1}), \eta_r^{a^1}(T^{f-2}), \dots, \eta_r^{a^{f-1}}(T), \eta_r(I)] \begin{matrix} x^0 \\ \vdots \\ x^{r-1} \end{matrix} \quad (2.8)$$

For example,

$$\eta_r(T) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ g_{r-1} \end{bmatrix}, \quad \eta_r(I) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \zeta_1(I) \\ \zeta_1(T) \\ \vdots \\ \zeta_1(T^{f-1}) \end{bmatrix}$$

Since the relation that $r \geq f$ is generally assumed, then

$$\begin{aligned} \zeta_1(I) &= [100 \dots 00] \\ \zeta_1(T) &= [010 \dots 00] \\ &\vdots \\ \zeta_1(T^{f-1}) &= [00 \dots 010 \dots 0] \end{aligned} \Rightarrow R = [I_{f \times f} \quad \Phi_{f \times (r-f)}] \quad (2.9)$$

where $\Phi = [0]_{f \times (r-f)}$ is a null matrix containing all zero elements.

Moreover, the relation that r is greater or equal to f makes all entries of the Q matrix become 0, i.e.,

$$Q = \Phi_{f \times f} \quad (2.10)$$

Example 2.1: Given a serial SRSG as shown in Figure 2.4, find its 3-channel analogy SRSG circuit.

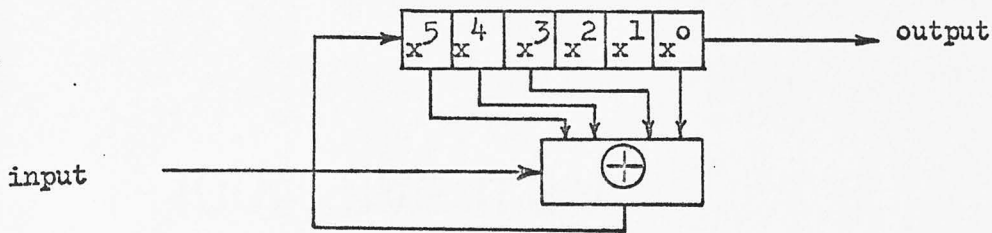


Figure 2.4 $g(x) = 1 + x^3 + x^4 + x^5 + x^6$

Solution: the set of characterizing matrices for the circuit of Figure 2.4 are given as follows:

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = [100000]$$

$$D = 0$$

$$T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad T^3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then, from Equation (2.3.a), we have

$$\begin{aligned}
 \begin{bmatrix} x_{t+1}^0 \\ x_{t+1}^1 \\ x_{t+1}^2 \\ x_{t+1}^3 \\ x_{t+1}^4 \\ x_{t+1}^5 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_t^0 \\ x_t^1 \\ x_t^2 \\ x_t^3 \\ x_t^4 \\ x_t^5 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_t^0 \\ a_t^1 \\ a_t^2 \end{bmatrix} \\
 &= \begin{bmatrix} x_t^3 \\ x_t^4 \\ x_t^5 \\ x_t^0 + x_t^3 + x_t^4 + x_t^5 + a_t^0 \\ x_t^0 + x_t^1 + x_t^3 + a_t^0 + a_t^1 \\ x_t^1 + x_t^2 + x_t^4 + a_t^1 + a_t^2 \end{bmatrix} \quad (2.11)
 \end{aligned}$$

Next, from Equation (2.3.b),

$$\begin{bmatrix} b_t^0 \\ b_t^1 \\ b_t^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_t^0 \\ x_t^1 \\ x_t^2 \\ x_t^3 \\ x_t^4 \\ x_t^5 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_t^0 \\ a_t^1 \\ a_t^2 \end{bmatrix} = \begin{bmatrix} x_t^0 \\ x_t^1 \\ x_t^2 \end{bmatrix} \quad (2.12)$$

The implementation of Equations (2.11) and (2.12) results in the 3-channel analogy SRSG as shown in Figure 2.5.

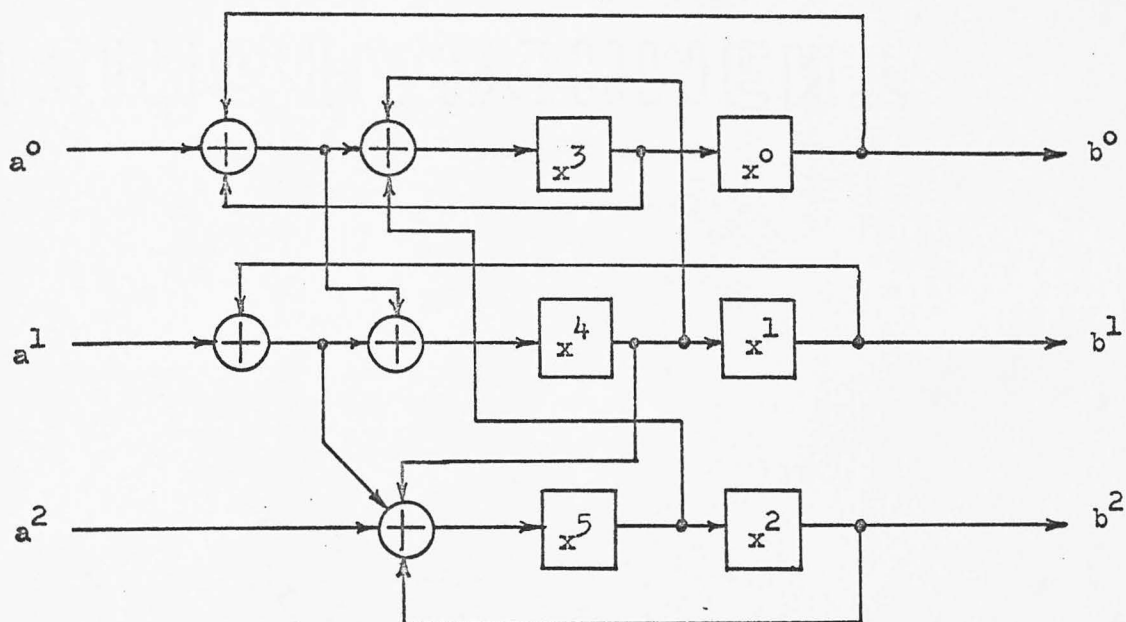


Figure 2.5 The 3-Channel Analogy SRSF for $g(x) = 1 + x^3 + x^4 + x^5 + x^6$

The matrix method is very formidable. Sometimes, it may be easier to use a direct approach. The method described below is similarly done by Hsiao and Sih^[22] for the f-channel analogy GF divider. The method is best described by the following example.

Example 2.2: Same example as the Example 2.1.

$$g(x) = 1 + x^3 + x^4 + x^5 + x^6, \quad f = 3$$

Solution: List the state transition Table 2.1.

Table 2.1

t	Input	x^5	x^4	x^3	$x^2 \ x^1 \ x^0$	Output
0	0	x_0^5	x_0^4	x_0^3	$x_0^2 \ x_0^1 \ x_0^0$	—
1	a_0^0	$a_0^0 + x_0^0 + x_0^3 + x_0^4 + x_0^5$	x_0^5	x_0^4	$x_0^3 \ x_0^2 \ x_0^1$	x_0^0
2	a_0^1	$a_0^1 + a_0^0 + x_0^0 + x_0^1 + x_0^3$	$a_0^0 + x_0^0 + x_0^3 + x_0^4 + x_0^5$	x_0^5	$x_0^4 \ x_0^3 \ x_0^2$	x_0^1
3	a_0^2	$a_0^0 + a_0^2 + x_0^2 + x_0^1 + x_0^4$	$a_0^1 + a_0^0 + x_0^0 + x_0^1 + x_0^3$	$a_0^0 + x_0^0 + x_0^3 + x_0^4 + x_0^5$	$x_0^5 \ x_0^4 \ x_0^3$	x_0^2

Note that $a_0^1 = a_1^0$, $a_0^2 = a_2^0$

Table 2.1 will not be completed until $t = f = 3$. The last row of $t = 3$ specifies the interconnections between all inputs, output, and storage devices. Therefore, we have, directly from Table 2.1,

$$b_t^0 = x_t^0$$

$$b_t^1 = x_t^1$$

$$b_t^2 = x_t^2$$

$$x_{t+1}^0 = x_t^3$$

$$x_{t+1}^1 = x_t^4$$

$$x_{t+1}^2 = x_t^5$$

$$x_{t+1}^3 = a_t^0 + x_t^0 + x_t^3 + x_t^4 + x_t^5$$

$$x_{t+1}^4 = a_t^0 + a_t^1 + x_t^0 + x_t^1 + x_t^3$$

$$x_{t+1}^5 = a_t^0 + a_t^1 + x_t^1 + x_t^2 + x_t^4$$

The above set of equations will result in the same circuit of Figure 2.5.

2.4 The parallel LFSR as a Galois field (GF) divider

In the modern algebraic coding theory, a cyclic code is defined in

terms of the generator polynomial $g(x)$. A codeword $f(x)$ is legal if and only if it is divisible by the $g(x)$. Since the polynomials form a polynomial ring defined over $GF(2)$, the implementation of $g(x)$ as a GF divider is the central part of the whole encoding and decoding system. The detailed discussion on the general algebraic coding theory is given in Peterson. [3] In this section, the matrix method of transforming a serial GF divider into its f-channel analogy GF divider is given. For the same input polynomial $v(x)$, the f-channel analogy GF divider will produce the same quotient polynomial $g(x)$, the remainder polynomial $r(x)$ as that of the serial GF divider, i.e.,

$$v(x) = g(x) g(x) + r(x) \quad (2.13)$$

With some modifications, the characterizing matrix equations of a GF divider can be written in the same form as Equations (2.1) and (2.2). The set of characterizing equations of the serial GF divider, defined according to Figure 1.2, are given as follows:

$$S_{t+1} = S_t T + U_t C \quad (2.14.a)$$

$$Y_t = S_t B + U_t D \quad (2.14.b)$$

where the matrices C , B , and D are specified by Equation (2.7). In the case of the f-channel analogy GF divider, we have

$$S_{t+1} = S_t T^f + U_t G \quad (2.15.a)$$

$$Y_t = S_t J + U_t Q' \quad (2.15.b)$$

where G is the input position matrix

$$G = \begin{bmatrix} \varphi_1(T^{f-1}) \\ \varphi_1(T^{f-2}) \\ \vdots \\ \varphi_1(I) \end{bmatrix}_{f \times r} \quad (2.16.a)$$

and J is the output position matrix.

$$J = [\eta_r(I), \eta_r(T), \dots, \eta_r(T^{f-2}), \eta_r(T^{f-1})]_{r \times f} \quad (2.16.b)$$

Q' is again a null matrix. Note that in Equation (2.15.a) and (2.15.b) the same T^f is used which means that the calculation of T^f need only be done once. However, the connection of the parallel GF divider specified from the T^f matrix is interrupted column-wise instead of row-wise as in the previous case. The following example demonstrates the above statements.

Example 2.3: Given $g(x) = x^3 + x^4 + x^5 + x^6$, find its 3-channel analogy GF divider.

Solution: In the serial case, we have

$$T = \begin{matrix} & \begin{matrix} x_t^0 & x_{t+1}^1 & x_{t+1}^2 & x_{t+1}^3 & x_{t+1}^4 & x_{t+1}^5 \end{matrix} \\ \begin{matrix} x_t^0 \\ x_t^1 \\ x_t^2 \\ x_t^3 \\ x_t^4 \\ x_t^5 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix} \quad C = [1 \ 0 \ 0 \ 0 \ 0 \ 0], \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Reading the T matrix column-wise, we have, e.g.,

$$x_{t+1}^3 = x_t^2 + x_t^5$$

The GF serial divider is shown in Figure 2.6.

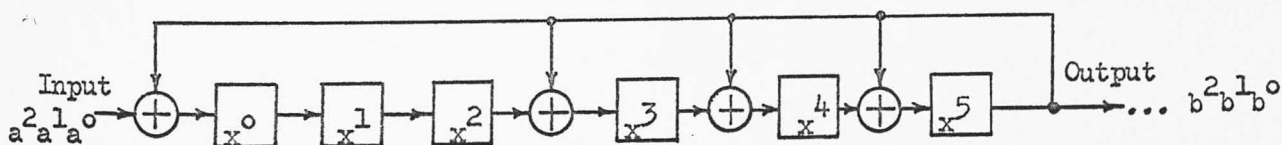


Figure 2.6 GF Divider $g(x) = 1 + x^3 + x^4 + x^5 + x^6$, $f=1$

The same T^2 and T^3 can be used here from previous Example 2.1 to find the implementation equations of the 3-channel analogy circuit.

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Therefore,

$$\begin{aligned} \begin{bmatrix} x_{t+1}^0 & x_{t+1}^1 & x_{t+1}^2 & x_{t+1}^3 & x_{t+1}^4 & x_{t+1}^5 \end{bmatrix} &= \begin{bmatrix} x_t^0 & x_t^1 & x_t^2 & x_t^3 & x_t^4 & x_t^5 \end{bmatrix} \begin{bmatrix} 000100 \\ 000010 \\ 000001 \\ 100111 \\ 110100 \\ 011010 \end{bmatrix} + \begin{bmatrix} a_t^0 & a_t^1 & a_t^2 \end{bmatrix} \begin{bmatrix} 001000 \\ 010000 \\ 100000 \end{bmatrix} \\ &= \left[a_t^2 + x_t^3 + x_t^4, a_t^1 + x_t^4 + x_t^5, a_t^0 + x_t^5, x_t^0 \right. \\ &\quad \left. + x_t^3 + x_t^4, x_t^1 + x_t^3 + x_t^5, x_t^2 + x_t^3 \right] \quad (2.17) \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} b_t^0 & b_t^1 & b_t^2 \end{bmatrix} &= \begin{bmatrix} x_t^0 & x_t^1 & x_t^2 & x_t^3 & x_t^4 & x_t^5 \end{bmatrix} \begin{bmatrix} 000 \\ 000 \\ 000 \\ 001 \\ 011 \\ 110 \end{bmatrix} + \begin{bmatrix} a_t^0 & a_t^1 & a_t^2 \end{bmatrix} \begin{bmatrix} 000 \\ 000 \\ 000 \end{bmatrix} \\ &= \left[x_t^5, x_t^4 + x_t^5, x_t^3 + x_t^4 \right] \quad (2.18) \end{aligned}$$

Equations (2.17) and (2.18) result in the circuit of Figure 2.7.

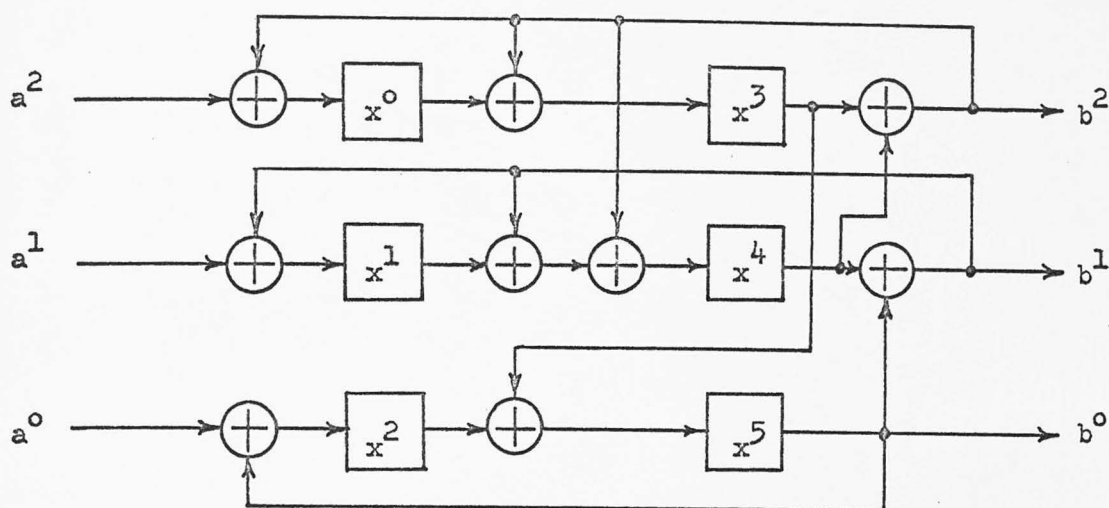


Figure 2.7 GF Divider $g(x) = 1 + x^3 + x^4 + x^5 + x^6$, $f=3$

2.5 The polynomial root approach to the transformation problem

As shown by Peterson, [3] successive shifts of a serial GF divider specified by $g(x)$ will give representations of successive powers of α — a root of $g(x)$. Therefore, to transform a serial GF divider into its f -channel analogy circuit is equivalent to calculating the set of coefficients of $r(\alpha^f)$ instead of $r(\alpha)$, i.e., to obtain a device that calculates the f^{th} power of the field element. This is illustrated through the same Example 2.3.

Let $f = 3$ and $r(\alpha) = d_0 + d_1\alpha + d_2\alpha^2 + d_3\alpha^3 + d_4\alpha^4 + d_5\alpha^5$,

$$\begin{aligned}
 \text{then } r(\alpha^3) &= \alpha^3(d_0 + d_1\alpha + d_2\alpha^2 + d_3\alpha^3 + d_4\alpha^4 + d_5\alpha^5) \\
 &= d_0\alpha^3 + d_1\alpha^4 + d_2\alpha^5 + d_3(1 + \alpha^3 + \alpha^4 + \alpha^5) + d_4(1 + \alpha + \alpha^3) \\
 &\quad + d_5(\alpha + \alpha^2 + \alpha^4) \\
 &= (d_3 + d_4) + (d_4 + d_5)\alpha + d_5\alpha^2 + (d_0 + d_3 + d_4)\alpha^3 \\
 &\quad + (d_1 + d_3 + d_5)\alpha^4 + (d_2 + d_3)\alpha^5 \\
 &= h_0 + h_1\alpha + h_2\alpha^2 + h_3\alpha^3 + h_4\alpha^4 + h_5\alpha^5
 \end{aligned} \tag{2.19}$$

The last Equation (2.19) gives a new set of coefficients,

$$\begin{aligned}
 h_0 &= d_3 + d_4 & h_1 &= d_4 + d_5 \\
 h_2 &= d_5 & h_3 &= d_0 + d_3 + d_4 \\
 h_4 &= d_1 + d_3 + d_5 & h_5 &= d_2 + d_3
 \end{aligned}$$

To express this in matrix notation, we have

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \end{bmatrix} = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 & d_4 & d_5 \end{bmatrix} \begin{bmatrix} 000100 \\ 000010 \\ 000001 \\ 100111 \\ 110100 \\ 011010 \end{bmatrix}$$

This transformation matrix is the same as T^3 , as previously calculated. It is noted in previous sections, the matrix T^f plays a key role to the serial-to-parallel transformation problem. In general, the relation between the matrix T and the root α of $g(x)$ and the residue class $\{x\}^*$ is very interesting. They can replace each other in many places without changing the general statement. Some examples are given below.

1. The period of a serial LFSR can be defined as follows,
Period $p \Rightarrow T^p = I$, or equivalently $\alpha^p = 1$
2. The minimum polynomial $m(\alpha)$ of α has the same form as the characteristic polynomial $\phi(T)$ in terms of the matrix T .
3. The r rows of T^j ($j = 1, 2, \dots, 2^r - 2$) is the following matrix

$$T^j = \begin{bmatrix} \alpha^j \\ \alpha^{j+1} \\ \vdots \\ \alpha^{j+r-1} \end{bmatrix} \quad (2.20)$$

The relation of 1 and 2 are well known. The relation 3 is an easy way of finding the transformation matrix T^f which will be treated in detail in Chapter 4.

* The substitution of α for $\{x\}$ and $\{x\}$ for α was stated in Peterson. [3]

CHAPTER 3

PARALLEL LFSR AS A COMPLETE SEQUENTIAL MACHINE

3.1 Basic definitions

The definitions and terminologies used in this chapter may be referred to Gill^[31] or Ginsburg.^[32] A LFSR is considered as a linear sequential machine, regardless whether it is single or multiple channel. This machine is complete, finite memory, and finite state. It is strongly connected since there always exists a path leading from state S_i to S_j for all $0 \leq i, j \leq 2^r - 1$.

Definition 3.1: A complete sequential machine is defined as a quintuple (S, U, Y, δ, Ψ) where

S is a nonempty finite set of "states"

U is a nonempty finite set of "inputs"

Y is a nonempty finite set of "outputs"

δ is a function (called the "next-state" function) which maps $S \times U$ into S

Ψ is a function (called the output function) which maps $S \times U$ into Y

The next-state function δ and the output function Ψ are characterized by the following two sets of matrix equations as shown in Sections 2.3 and 2.4.

$$\delta(S, U) = S'_{t+1} = T^f S'_t + L U'_t$$

$$\Psi(S, U) = Y'_t = R S'_t + Q U'_t$$

$$\delta(S, U) = S_{t+1} = S_t T^f + U_t G$$

$$\Psi(S, U) = Y_t = S_t J + U_t Q'$$

For any LFSR of r -stages, the set S contains 2^r states $0, 1, 2, \dots, 2^r - 1$. The input set U and the output set Y are defined as follows.

Since the parallel LFSR has f -channel input/output simultaneously, it is necessary to select a set of symbols to represent all 2^f binary patterns. Let the 2^f binary patterns be the elements α^i for all $i = 0, \dots, 2^f - 2$ of $GF(2^f)$ where α is a primitive root of $GF(2^f)$. We then have,

$$U = \{\phi, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^f-2}\} \quad (3.1.a)$$

$$Y = \{\phi, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^f-2}\} \quad (3.1.b)$$

Appendix IA lists a table of all α 's for $f = 2, 3, 4, 5, 6$. The method of constructing this table may be referred to Peterson. [3]

Example 3.1: Given $g(x) = 1 + x + x^2$ and $f = 2$, construct its transition table and state diagram.

Solution:

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad T^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

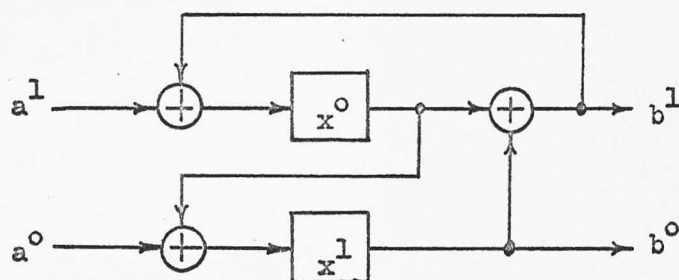


Figure 3.1 A GF Divider

Since $f = 2$, we have,

$$S = \{(x^0, x^1) \mid x^0, x^1 \in \text{GF}(2) \Rightarrow (0,0), (0,1), (1,0), (1,1)\}$$

$$U = \{(a^1, a^0) \mid a^1, a^0 \in \text{GF}(2) \Rightarrow \phi=(0,0), \alpha^0 = (1,0), \alpha^1 = (0,1), \alpha^2 = (1,1)\}$$

$$Y = \{(b^1, b^0) \mid b^1, b^0 \in \text{GF}(2) \Rightarrow \phi=(0,0), \alpha^0 = (1,0), \alpha^1 = (0,1), \alpha^2 = (1,1)\}$$

$$SXU \xrightarrow{\delta} S \Rightarrow x_{t+1}^0 = a_t^1 + x_t^0 + x_t^1$$

$$x_{t+1}^1 = a_t^0 + x_t^0$$

$$SXU \xrightarrow{\Psi} S \Rightarrow b_t^1 = x_1^0 + x_t^1$$

$$b_t^0 = x_t^1$$

With the finite state machine completely specified, we have the following transition Table 3.1

Table 3.1

		Y				S			
S	U	ϕ	α^0	α^1	α^2	ϕ	α^0	α^1	α^2
		ϕ	ϕ	ϕ	ϕ	0	2	1	3
0 (0,0)		ϕ	ϕ	ϕ	ϕ	0	2	1	3
1 (0,1)		α^2	α^2	α^2	α^2	2	0	3	1
2 (1,0)		α^0	α^0	α^0	α^0	3	1	2	0
3 (1,1)		α^1	α^1	α^1	α^1	1	3	0	2

The state diagram is shown in Figure 3.2 which is a Moore machine. [31]

The input alphabet or the output alphabet form a semi-group under concatenation.

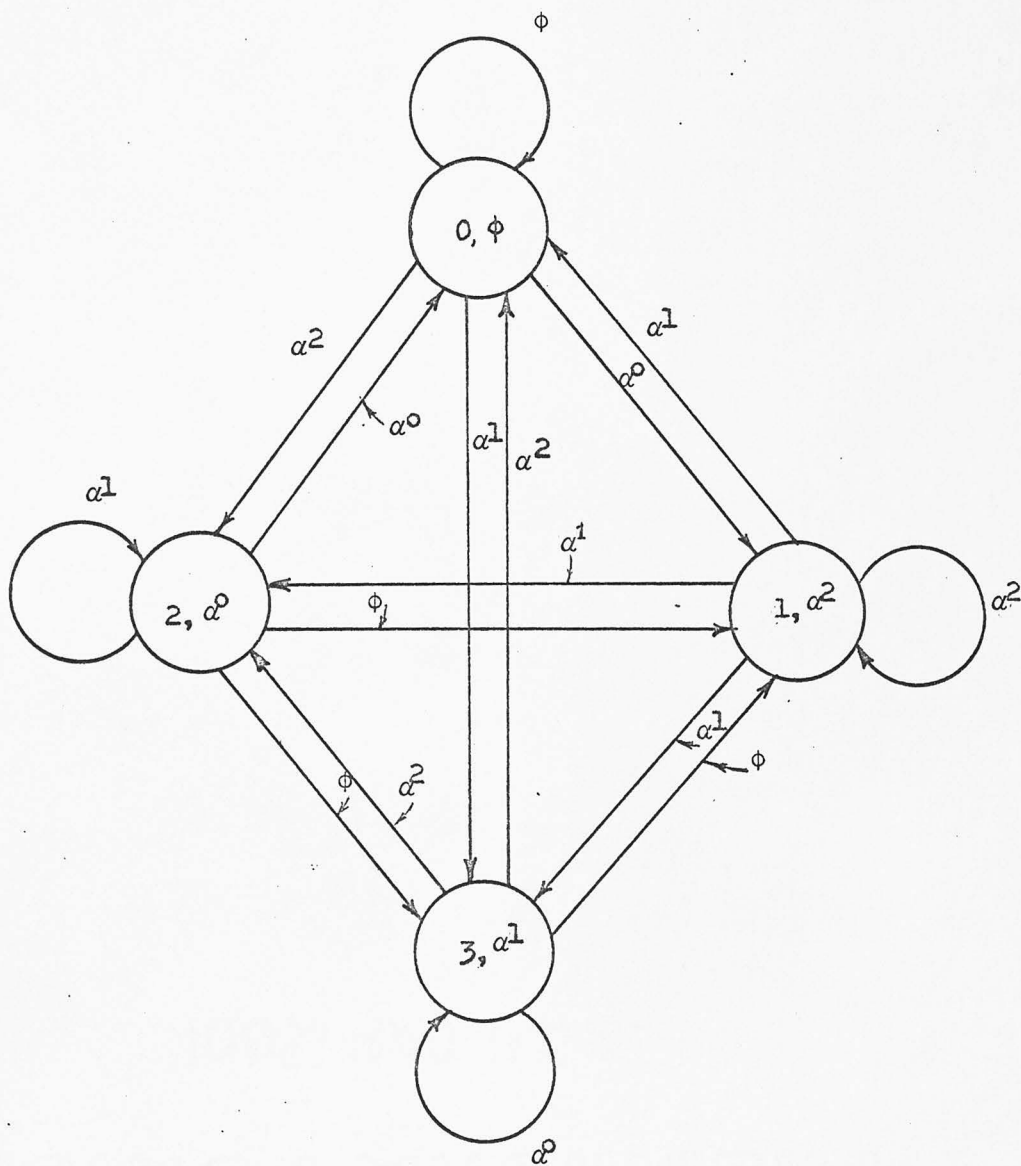


Figure 3.2 State Diagram for
 $g(x) = 1 + x + x^2$ and $f=2$

Definition 3.2: Let $U = \{\phi, \alpha^0, \alpha^1, \dots, \alpha^{2^f-2}\} \in GF(2^f)$ denote the input alphabet. The input dictionary, denoted by U^* , is defined to be the set of all words formed from U .

U^* contains the word with no letters, the null word, which will be denoted by Λ .

Definition 3.3: For $x, y \in U^*$, the concatenation of x and y is written xy and is defined by writing x followed by y .

Clearly, $xy \neq yx$

But $\Lambda x = x\Lambda = x$ for all $x \in U^*$

Since $(xy)z = x(yz)$ for any $x, y, z \in U^*$, this implies that concatenation is an associative operation and the parentheses are generally not needed in grouping words.

Theorem 3.1: $\langle U^*, \cdot, \Lambda \rangle$ is the free monoid (semi-group with identity) on the 2^f-1 generators $\phi, \alpha^0, \alpha^1, \dots, \alpha^{2^f-2}$.

Proof: The proof is obvious since the closure property is clearly satisfied and the associative law is also fulfilled by the concatenation operation.

Here the identity element is Λ since

$$\Lambda x = x\Lambda = x$$

Q.E.D.

3.2 The equivalent machine and similar transformation

In this section, an equivalent machine can be obtained due to a similar transformation. The advantage of establishing equivalent machines is to obtain a more economic machine than a given machine, which will perform the same kind of operation as the original machine. In the case of serial LFSR, a minimum machine was obtained by Roth. [33]

Definition 3.4 [32] [31]: Two complete sequential machines are called equivalent if both do the same work, i.e., both machines perform the same

transformation from the set of input sequences to the set of output sequences.

Two LFSR are said to be equivalent if and only if they "reduce" or "process" the same data so as to yield identical results. [32]

Definition 3.5: Two autonomous parallel LFSR M_1 and M_2 characterized by two rxr matrices T_1 and T_2 are said to be similar if there exists a nonsingular rxr matrix A with elements over $GF(2)$ such that

$$T_1 = A T_2 A^{-1} \quad (3.2.a)$$

the similar transformation exists. Conversely, we say M_2 is similar to M_1 if

$$T_2 = A^{-1} T_1 A \quad (3.2.b)$$

Since the relation of similarity is symmetric, it suffices to say merely that M_1 and M_2 are similar and denotes it by $M_1 \sim M_2$. There are several properties of a similar transformation which makes it useful in establishing equivalent machines.

1. If $M_1 \sim M_2$, then their characteristic equations have the same invariant factors, i.e., the polynomials $|T_1 - \lambda I|$ and $|T_2 - \lambda I|$ are equivalent over $GF_2[x]$.⁺

Proof: see Perlis,^[34] p. 143.

2. If M_1 is similarly transformed into M_2 , then they have the same cycle set structure, i.e., the dimension of the subspace and the length of its cycles are the same.

Proof: see Carter,^[35] p. 3.

In fact, two similar matrices always have the same characteristic determinant

$$|T_1| = |A^{-1}| |T_2| |A| = |T_2|$$

and the relation $|T_1| = |T_2|$ is a necessary but not sufficient condition for the similarity of the matrices T_1 and T_2 . This is

⁺ $GF_2[x]$ is the polynomial domain containing all polynomials in x with coefficients in $GF(2)$.

The above three properties have established the result of a similar transformation.

Since $M_1 \sim M_2$ and $T_1 \neq T_2$ which implies T_1 may have less number of 1-entry in the matrix or vice versa, we can always select the machine with less number of 1-entry in its T matrix because the number of exclusive-or gates used in the circuit is directly associated with the number of 1-entry in the matrix.

One immediate question is how to find the transformation matrix A . There is no definite answer to this question. One type of the A matrix is the following.

$$A = [\eta_r(I), \eta_r(T), \eta_r(T^2), \dots, \eta_r(T^{r-1})] \quad (3.4)$$

Unfortunately, the above A matrix does not always give a similar matrix of less number of 1-entry.

Next, the input and output connection matrix of a similar machine is derived as follows.

(i) SRSG: Let $V = A^{-1} T^f A$, then $T^f = AVA^{-1}$. Substituting it in Equation (2.3.a), we have

$$S'_{t+1} = AVA^{-1} S'_t + LU'_t$$

$$Y'_t = RS'_t + QU'_t$$

Let $\tilde{S}'_t = A^{-1} S'_t$, then $\tilde{S}'_{t+1} = A^{-1} S'_{t+1}$

And $\tilde{AS}'_{t+1} = AVS'_t + L U'_t$

$$\tilde{S}'_{t+1} = \tilde{V}S'_t + (A^{-1} L) U'_t \quad (3.5.a)$$

$$Y'_t = (RA) \tilde{S}'_t + Q U'_t \quad (3.5.b)$$

therefore, the connection matrices are read as follows.

$$\tilde{L} = A^{-1}L, \quad \tilde{R} = RA$$

(ii) GF divider: Let $V_1 = AT^f A^{-1}$, then $T^f = A^{-1}V_1A$. Substituting it in Equation (2.15.a), we have

$$S_{t+1} = S_t A^{-1} V_1 A + U_t G$$

$$Y_t = S_t J + U_t Q'$$

Let $\tilde{S}_t = S_t A^{-1}$, then $\tilde{S}_{t+1} = S_{t+1} A^{-1}$

And $\tilde{S}_{t+1} = \tilde{S}_t V_1 + U_t (GA^{-1})$ (3.6.a)

$$Y_t = \tilde{S}_t (AJ) + U_t Q' \quad (3.6.b)$$

Therefore the connection matrices are read as follows.

$$\tilde{G} = G A^{-1}, \quad \tilde{J} = A J$$

Example 3.2: Given $g(x) = 1 + x + x^4$ and $f = 4$ implemented as a GF divider, find an equivalent circuit.

$$T^4 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

A is chosen as follows.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow A^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$V = A T^4 A^{-1} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\det |V - \lambda I| = \lambda^4 + \lambda + 1$$

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad J = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

then

$$\tilde{G} = GA^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \tilde{J} = AJ = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Without any simplification, the original circuit and its similar circuit are implemented as shown in Figures 3.3 and 3.4 respectively.

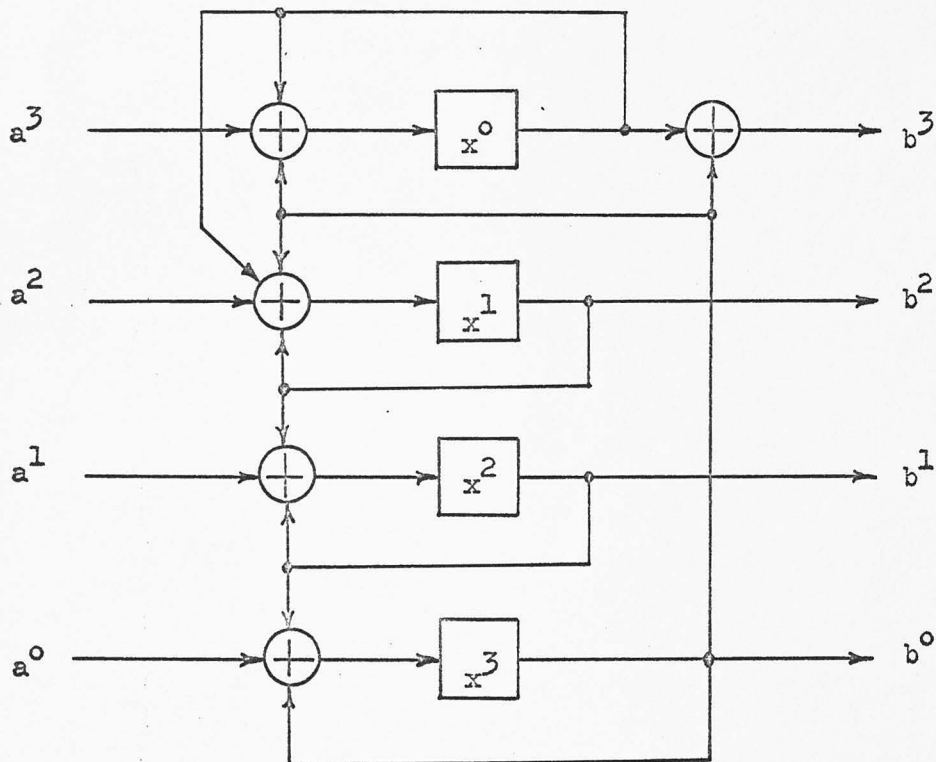


Figure 3.3

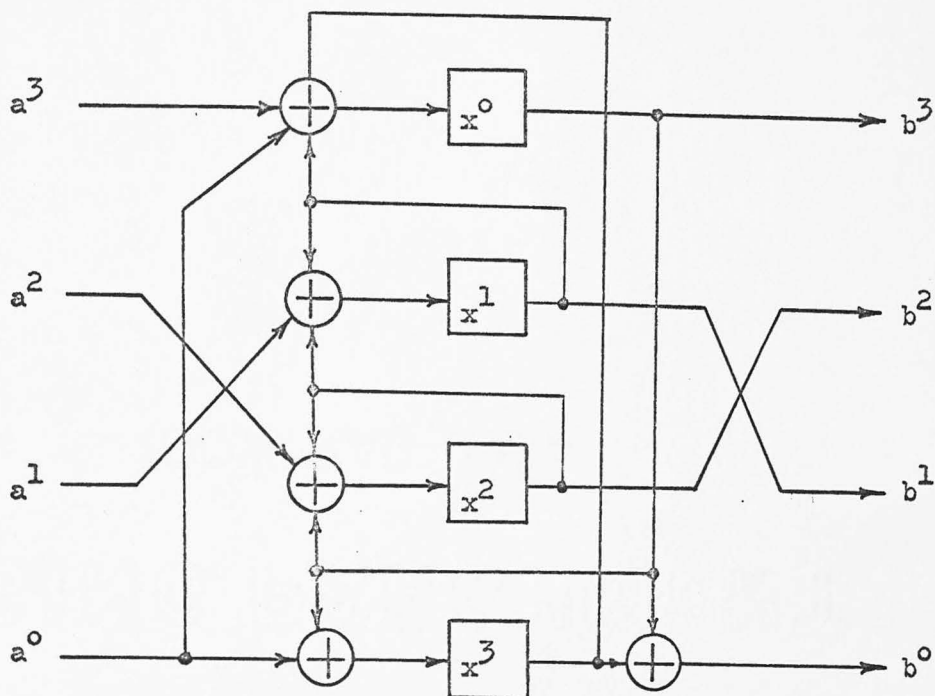


Figure 3.4

Comparing these two circuits, we note that the similar machine uses one exclusive-or gate less than the original circuit.

Another class of similar transformation matrix may be tried for minimization. This class of matrix is involuntary, [36] i.e.,

$$A_n A_n = I, \text{ or } A_n = A_n^{-1}$$

where

$$A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\vdots$$

$$A_{2n} = \begin{bmatrix} A_n & A_n \\ 0 & A_n \end{bmatrix}$$

The order of A_n for all n equals to 2^i for $i = 1, 2, \dots$ and therefore the application is much limited.

CHAPTER 4

ON PERIODS AND CHARACTERISTIC EQUATIONS OF THE PARALLEL LFSR

4.1 On periods of the parallel LFSR

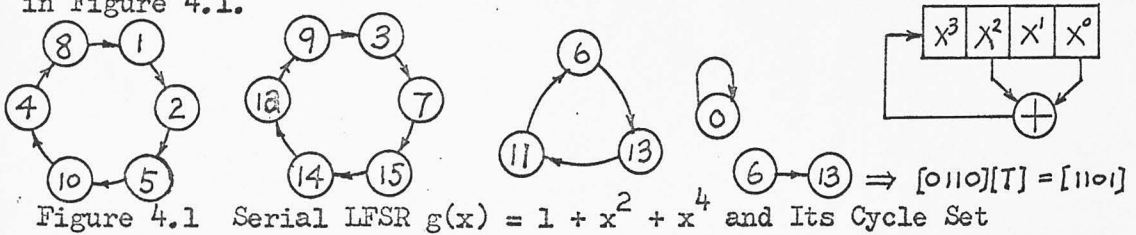
The periods of a f-channel analogy parallel LFSR, denoted by P_p , can be easily obtained from the periods of the original serial LFSR, denoted by P_s , using a set of converting rules. Since any serial LFSR has more than one isolated state transition path, it has at least two cycles which forms a cycle set. The number of states in each closed circular path is defined as the period of this cycle. All cycles in the state diagram have no branch state.

Definition 4.1: The set of all possible cycles is called the cycle set of the LFSR of the companion matrix T .

Notation: Cycle set = $\{\mu_1(k_1), \dots, \mu_n(k_n)\}$ where $\mu_i(k_i)$ implies μ_i cycles of length k_i .

Definition 4.2: The period P_s of a serial LFSR is defined as the smallest integer for which $T^{P_s} = I$, where I is the identity matrix.

Example 4.1: $g(x) = 1 + x^2 + x^4$, cycle set = 1(1), 1(3), 2(6) is shown in Figure 4.1.



Extensive study has been made on the periods of the serial LFSR on the structure of the characteristic polynomial $g(x)$ of the circuit. [10][37][12] Therefore we shall assume that the periods of a given serial LFSR are known and try to establish the link between each P_s and its corresponding P_p . The result deals with one cycle; and since all cycles are isolated to each

other, that result can certainly be applied to other cycles as well.

The main difference between the serial and the parallel LFSR lies on the sequence of state transition. If P_s is the period of a serial LFSR, then its states transition sequence with an initial state S_1 is given as follows.

$$S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_{P_s} \rightarrow S_1 \rightarrow \dots$$

If the parallel LFSR is designed to advance f states per shift, then the state transition becomes

$$S_1 \rightarrow S_{f+1} \rightarrow S_{2f+1} \rightarrow \dots \rightarrow S_{mf+1} \rightarrow \dots$$

Example 4.2: $g(x) = 1 + x^2 + x^4$, and $f = 2$, find its cycle set based on the result of Example 4.1?

The new cycle set is shown in Figure 4.2.

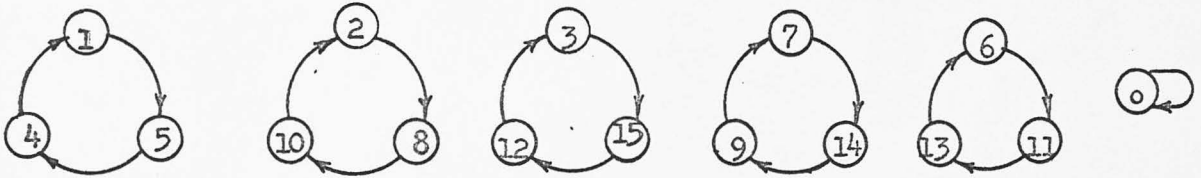


Figure 4.2 Cycle Set = $\{1(1), 5(3)\}$

In the parallel LFSR, each cycle has a period less than or equal to that of the original serial LFSR. This is quite obvious since no state can be met twice before a cycle is completed. Therefore, the following three theorems are given.

Theorem 4.1: If $P_s = mf$, then $P_p = m$ where m is a positive integer.

Proof: By definition 4.2,

$$T^{P_s} = I$$

$$\text{then } P_s = mf \Rightarrow T^{mf} = I \Rightarrow [T^f]^m = I$$

again by definition 4.2,

$$[T^f]^P_p = I$$

$$\therefore m = P_p$$

Theorem 4.2: If f and P_s are relative prime, then $P_p = P_s$.

Proof: $T^S = I$ (4.1)

and $[T^f]^P_s = I$ (4.2)

Comparing Equations (4.1) and (4.2), with the fact that $I^k = I$ where k is a positive integer, we have,

$$f P_p = k P_s$$

or $P_p = k \frac{P_s}{f}$ (4.3)

Since P_s and f are relative prime, then the smallest k that satisfies Equation (4.3) is " k equal to f " for which P_p is an integer. Therefore, we have

$$P_p = P_s$$

Theorem 4.3: If f and P_s have a common factor $c > 1$, i.e., $f = c a$ and

$P_s = c b$ where a and b are positive integers and relative prime,

then $P_p = \frac{P_s}{c}$

Proof: From the proof of Theorem 4.2, we have

$$P_p = k \frac{P_s}{f} = k \frac{cb}{ca} \quad (4.4)$$

The smallest integer k to make P_p an integer in Equation (4.4) is " k equal to a ," Therefore, we have

$$P_p = \frac{P_s}{c}$$

The above three theorems have established the link between one period of a given serial LFSR to the corresponding period of its f -channel analogy parallel LFSR. Therefore all periods of a parallel LFSR can be found on the basis of the known periods of its original serial LFSR.

4.2 Methods of finding the characteristic equation of an autonomous f-channel analogy parallel LFSR

The cycle set structure of an autonomous serial LFSR is completely specified by its characteristic polynomial.

Definition 4.3: The characteristic equation $\phi_f(\lambda) = 0$ of a $r \times r$ matrix T^f is the determinant $|T^f - \lambda I| = 0$, where the determinant $|T^f - \lambda I| = \phi_f(\lambda)$ is the characteristic polynomial.

Definition 4.4: The minimal polynomial of T is the monic polynomial $m(\lambda)$ (coefficient of the highest power term equals to 1) of lowest degree such that $m(T) = 0$.

Therefore, $m_f(\lambda) | \phi_f(\lambda)$ and in general, $m(\lambda) \neq \phi(\lambda)$

Theorem 4.4: In the serial case, $f = 1$, the characteristic and minimum polynomial of the companion matrix of $g(x)$ are both equal to $g(x)$, i.e.,

$$g(\lambda) = \phi_1(\lambda) = m_1(\lambda)$$

Proof: see p. 148 of Perlis. [34]

The general procedure of finding $\phi_f(\lambda)$ of an autonomous f-channel analogy parallel LFSR is diagrammed as follows.

$$g(x) \rightarrow T \rightarrow T^f \rightarrow \phi_f(\lambda)$$

Various methods of finding $\phi_f(\lambda)$ are discussed below.

(i) Direct method: By Definition 4.3,

$$\phi_f(\lambda) = |T^f - \lambda I| \quad (4.5)$$

We shall first find T^f and then evaluate the determinant $|T^f - \lambda I|$ directly.

Theorem 4.5:[38] (Cayley-Hamilton) Every square matrix satisfies its own characteristic polynomial, i.e.,

$$\phi_f(T^f) = 0 \quad (4.6)$$

Example 4.3: Given $g(x) = 1+x+x^4$, find $\phi_2(\lambda)$ and $\phi_3(\lambda)$

Solution:

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad T^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\phi_2(\lambda) = |T^2 - \lambda I| = \begin{vmatrix} \lambda & 0 & 1 & 0 \\ 0 & \lambda & 0 & 1 \\ 1 & 1 & \lambda & 0 \\ 0 & 1 & 1 & \lambda \end{vmatrix} = \lambda^4 + \lambda + 1$$

Try Cayley-Hamilton theorem,

$$T^8 + T^4 + I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \Phi \\ \end{bmatrix}_{4 \times 4}$$

Next,

$$\phi_3(\lambda) = |T^3 - \lambda I| = \begin{vmatrix} \lambda & 0 & 0 & 1 \\ 1 & 1+\lambda & 0 & 0 \\ 0 & 1 & 1+\lambda & 0 \\ 0 & 0 & 1 & 1+\lambda \end{vmatrix} = \lambda^4 + \lambda^3 + \lambda^2 + \lambda + 1$$

Try Cayley-Hamilton theorem again,

$$\begin{aligned} T^{12} + T^9 + T^6 + T^3 + I &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \\ &+ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \Phi \\ \end{bmatrix}_{4 \times 4} \end{aligned}$$

The direct method of finding a characteristic equation becomes difficult if the dimension of T^f is large.

Several alternative methods are given below mostly based on the work of Albert. [26]

(i) Polynomial root method: This method provides an easy way of finding T^j based on the table of all powers of a root α^+ of $g(x)$. Appendix 1 lists several tables for different $g(x)$.

Since all elements in $GF(2^r)$ form a r -dimensional vector space with a set of linear independent basis as $1, \alpha, \alpha^2, \dots, \alpha^{r-1}$ where $\alpha^i (i = 0, 1, 2, \dots, r-1)$ is a binary r -tuple defined as follows.

$$\begin{aligned}\alpha^0 &= \overbrace{1\ 0\ 0\ \dots\ 0\ 0}^r \\ \alpha^1 &= 0\ 1\ 0\ \dots\ 0\ 0 \\ &\vdots \\ \alpha^{r-1} &= 0\ 0\ 0\ \dots\ 0\ 1\end{aligned}$$

Then any element $\alpha^j \in GF(2^r)$ can be expressed as

$$\alpha^j = \xi_{11} + \xi_{12} \alpha + \xi_{13} \alpha^2 + \dots + \xi_{1r} \alpha^{r-1}$$

with $\xi_{ij} \in GF(2)$.

Next, compute the products $\alpha^{j+i-1} = \xi_{i1} + \xi_{i2} \alpha + \dots + \xi_{ir} \alpha^{r-1}$ for $i = 1, 2, 3, \dots, (r-1)$ and so obtain

$$T^j = \left[\xi_{ij} \right]_{r \times r} = \begin{bmatrix} \alpha^j \\ \alpha^{j+1} \\ \vdots \\ \alpha^{j+r-1} \end{bmatrix}_{r \times r} \quad (4.7)$$

+ If $g(x)$ has the period P_s , then $g(x) \mid x^{P_s} - 1$. The root α of $g(x)$ is then a P_s -th root of unity.

The result of Equation 4.7 can be verified through induction.

Let $k = 0$,

$$T^k = T^0 = I_{r \times r} = \begin{bmatrix} \alpha^0 & & \\ & \alpha^1 & \\ & & \ddots \\ & & & \alpha^{r-1} \end{bmatrix}$$

For $k = 1$,

$$T^k = T^1 = T = \begin{bmatrix} \alpha^1 & & \\ & \alpha^2 & \\ & & \ddots \\ & & & \alpha^r \end{bmatrix} \quad (4.8)$$

Since α is also equivalent to the residue class x as defined by Peterson,^[3] i.e.,

$$\alpha \equiv \{x\}$$

then, from the fact of $g_r = 1$,

$$\alpha^r \equiv g_0 \alpha^0 + g_1 \alpha^1 + \dots + g_{r-1} \alpha^{r-1} \quad \text{mod } g(\alpha)$$

or

$$\alpha^r = (g_0, g_1, \dots, g_{r-1}) \quad (4.9)$$

Substituting Equation (4.9) into Equation (4.8), we have exactly the companion matrix T . Next, we assume that the result holds for the case of $k = j$, i.e.,

$$T^j = \begin{bmatrix} \alpha^j & & \\ & \alpha^{j+1} & \\ & & \ddots \\ & & & \alpha^{j+r-1} \end{bmatrix}$$

Then, we shall show that it is true for $k = j+1$.

$$\begin{aligned}
 T^{j+1} = T \cdot T^j &= \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & & \dots & & 1 \\ g_0 & g_1 & & \dots & g_{r-1} & \end{bmatrix} \begin{bmatrix} \alpha^j \\ \alpha^{j+1} \\ \vdots \\ \alpha^{j+r-1} \end{bmatrix} \\
 &= \begin{bmatrix} \alpha^{j+1} \\ \alpha^{j+2} \\ \vdots \\ \alpha^{j+r-1} \\ g_0 \alpha^j + g_1 \alpha^{j+1} + \dots + g_{r-1} \alpha^{j+r-1} \end{bmatrix} \quad (4.10)
 \end{aligned}$$

But,

$$\begin{aligned}
 &g_0 \alpha^j + g_1 \alpha^{j+1} + \dots + g_{r-1} \alpha^{j+r-1} \\
 &= \alpha^j (g_0 + g_1 \alpha + \dots + g_{r-1} \alpha^{r-1}) \\
 &= \alpha^j \cdot \alpha^r \\
 &= \alpha^{j+r}
 \end{aligned}$$

Therefore, the last row of Equation (4.10) is exactly α^{r+j} and the proof is thus completed.

Example 4.4: Given $g(x) = 1 + x + x^4$ and let α be a root of $g(x)$, then any power of T can be found from the following Table 4.1, which lists all powers of α .

Table 4.1 All Powers of α

α^0	1 0 0 0
α^1	0 1 0 0
α^2	0 0 1 0
α^3	0 0 0 1
α^4	1 1 0 0
α^5	0 1 1 0
α^6	0 0 1 1
α^7	1 1 0 1
α^8	1 0 1 0
α^9	0 1 0 1
α^{10}	1 1 1 0
α^{11}	0 1 1 1
α^{12}	1 1 1 1
α^{13}	1 0 1 1
α^{14}	1 0 0 1

Then

$$T^j = \begin{bmatrix} \alpha^j \\ \alpha^{j+1} \\ \alpha^{j+2} \\ \alpha^{j+3} \end{bmatrix}$$

Suppose $j = 8$, then

$$T^8 = \begin{bmatrix} \alpha^8 \\ \alpha^9 \\ \alpha^{10} \\ \alpha^{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Next, we shall consider a reducible $g(x)$ with α as its root.

Example 4.5: Given $g(x) = 1 + x^2 + x^4$, then Table 4.2 can be constructed.

Table 4.2 All Powers of α for $g(x) = 1 + x^2 + x^4$

α^0	1 0 0 0
α^1	0 1 0 0
α^2	0 0 1 0
α^3	0 0 0 1
α^4	1 0 1 0
α^5	0 1 0 1

Therefore, from the above Table 4.2, we have,

$$T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad \& \quad T^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The above method can be used to find T^{-j} for all j . Let the period of the cycle which contains α be P_1 , then $\alpha^{P_1} = 1$. Therefore,

$$T^{P_1} = I \Rightarrow T^j T^{P_1-j} = I \Rightarrow T^{-j} = T^{P_1-j} \quad (4.11)$$

Note that Tables 4.1 and 4.2 are actually the same as the B_A matrix defined by Birdsall and Ristenbatt.^[11] However, they did not point out how T^j can be found from the B_A matrix. Furthermore, no word on polynomial root was mentioned in relation to the B_A matrix.

The third method to be described below utilizes tables in the Appendix C of Peterson^[3] and thus restricted itself to a certain class of polynomials.

(iii) Minimum polynomial method: Here the definition of minimum polynomial is given again based on the polynomial root rather than the T matrix.

Definition 4.5: Let β be any element of $\text{GF}(2^r)$, the monic polynomial $m(x)$ of smallest degree over $\text{GF}_2[x]$ such that $m(\beta) = 0$ is called the minimum polynomial.

If $g(x)$ is a primitive polynomial and α be its primitive root, then

$$\phi_f(\lambda) = [m_f(\lambda)]^C$$

where $m_f(\lambda)$ is the minimum polynomial of α^f with degree m and $r = mc$.

Proof: The proof is almost the same as that of Theorem 4.8 of Albert (P100). [26]

Theorem 3.31 of Albert implies that $\phi_f(\lambda)$ has coefficient in $\text{GF}(2)$ and $\phi_f(\alpha^f) = 0$. It follows that the minimum function over $\text{GF}(2)$ of α^f divides $\phi_f(\lambda)$. The result of $r = mc$ is given in the Corollary of Theorem 4.3 of Albert. The polynomial $m_f(x)$ has α_1 as a root and so is divisible by $g(x)$ since $\alpha_1, \dots, \alpha_r$ are r distinct roots of $g(x)$, i.e.,

$$g(x) = (x - \alpha_1) \dots (x - \alpha_r)$$

Then $m_f(\alpha_1) = 0$, and so every one of the roots of $\phi_f(\lambda)$ is a root of $m_f(\lambda) = 0$.

Now $\phi_f(\lambda) = m_f(\lambda)^C \phi_o(\lambda)$ where $\phi_o(\lambda)$ is either a non-constant polynomial prime to $m_f(\lambda)$ or $\phi_o(\lambda) = 1$.

Since $\phi_o(\lambda)$ must have a root in common with $m_f(\lambda)$, it is divisible by $m_f(\lambda)$, and the resulting contradiction implies that $\phi_o(\lambda) = 1$.

Example 4.5: Let α be a primitive root of $g(x) = 1 + x + x^4$, then from p.254 of Peterson, [3] we find

$$\begin{aligned} m_1(x) &= 1 + x + x^4 \\ m_3(x) &= 1 + x + x^2 + x^3 + x^4 \\ m_5(x) &= 1 + x + x^2 \end{aligned}$$

Therefore,

$$\phi_1(\lambda) = [m_1(\lambda)]^1 = 1 + \lambda + \lambda^4$$

$$\phi_3(\lambda) = [m_3(\lambda)]^1 = 1 + \lambda + \lambda^2 + \lambda^3 + \lambda^4$$

$$\phi_5(\lambda) = [m_5(\lambda)]^2 = (1 + \lambda + \lambda^2)^2 = 1 + \lambda^2 + \lambda^4$$

The above result can be easily verified through other methods. In $GF(2)$, as derived from Theorem 6.26 of Peterson,^[3] the following is true:

$$\begin{aligned} m_i(x) &= m_{2i}(x) = m_{2^2 i}(x) = \dots = m_{2^{j_i} \bmod (2^r - 1)}(x) = \dots \\ &= m_{2^{m-1} i \bmod (2^r - 1)}(x) \end{aligned}$$

Therefore in Example 4.5, we have

$$\phi_1(\lambda) = \phi_2(\lambda) = \phi_4(\lambda) = \phi_8(\lambda)$$

$$\phi_3(\lambda) = \phi_6(\lambda) = \phi_{12}(\lambda) = \phi_9(\lambda)$$

$$\phi_5(\lambda) = \phi_{10}(\lambda)$$

The last result can be stated in another way: If T is associated with $g(x)$ and $\phi(T) = 0$, then $\phi(T^2) = \phi(T^{2^2}) = \dots = \phi(T^{2^{m-1}}) = 0$

Definition 4.6: ^[11] A similarity class is a set of matrices $T^i, T^{2i}, T^{2^2 i}, \dots, T^{2^{m-1} i}$ such that they all satisfy the same characteristic equation. In

the above example, the matrices T, T^2, T^4, T^8 form a similarity class.

Example 4.6: Given $g(x) = 1 + x^2 + x^5$, we shall then have the following similarity classes.

$$\begin{aligned} \{ \alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16} \} &\Rightarrow m_1(x) = 1 + x^2 + x^5 \Rightarrow \phi_1(\lambda) = \phi_2(\lambda) = \phi_4(\lambda) \\ &= \phi_8(\lambda) = \phi_{16}(\lambda) = m_1(\lambda) \\ \{ \alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17} \} &\Rightarrow m_3(x) = 1 + x^2 + x^3 + x^4 + x^5 \Rightarrow \phi_3(\lambda) = \phi_6(\lambda) \\ &= \phi_{12}(\lambda) = \phi_{24}(\lambda) = \phi_{17}(\lambda) = m_3(\lambda) \\ \{ \alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^{18} \} &\Rightarrow m_5(x) = 1 + x + x^2 + x^3 + x^5 \Rightarrow \phi_5(\lambda) = \phi_{10}(\lambda) \\ &= \phi_{20}(\lambda) = \phi_9(\lambda) = \phi_{18}(\lambda) = m_5(\lambda) \end{aligned}$$

$$\{\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{25}, \alpha^{19}\} \Rightarrow m_7(x) = 1 + x + x^2 + x^4 + x^5 \Rightarrow \phi_7(\lambda) = \phi_{14}(\lambda) \\ = \phi_{28}(\lambda) = \phi_{25}(\lambda) = \phi_{19}(\lambda) = m_7(\lambda)$$

$$\{\alpha^{11}, \alpha^{22}, \alpha^{13}, \alpha^{26}, \alpha^{21}\} \Rightarrow m_{11}(x) = 1 + x + x^3 + x^4 + x^5 \Rightarrow \phi_{11}(\lambda) \\ = \phi_{22}(\lambda) = \phi_{13}(\lambda) = \phi_{26}(\lambda) = \phi_{21}(\lambda) = m_{11}(\lambda)$$

$$\{\alpha^{15}, \alpha^{30}, \alpha^{29}, \alpha^{27}, \alpha^{23}\} \Rightarrow m_{15}(x) = 1 + x^3 + x^5 \Rightarrow \phi_{15}(\lambda) = \phi_{30}(\lambda) = \phi_{29}(\lambda) \\ = \phi_{27}(\lambda) = \phi_{23}(\lambda) = m_{15}(\lambda)$$

This general method is actually closely related to the construction of $g(x)$ of a Bose-Chaudhuri-Hocquenghem code given that $\alpha_1, \alpha_2, \dots, \alpha_{2t-1}$ are roots of a code vector.

The similarity class has been related to the cyclotomic polynomial, i.e., the superscript numbers form a cyclotomic coset. For example, the $g(x)$ of Example 4.6 has $P_g = 31$, the multiplicative group consists of integers from 1 to 30. The decomposition into cosets is

$$C_0: 1 \ 2 \ 4 \ 8 \ 16$$

$$C_1: 3 \ 6 \ 12 \ 24 \ 17$$

$$C_2: 9 \ 18 \ 5 \ 10 \ 20$$

$$C_3: 27 \ 23 \ 15 \ 30 \ 29$$

$$C_4: 19 \ 7 \ 14 \ 28 \ 25$$

$$C_5: 26 \ 21 \ 11 \ 22 \ 13$$

Many theorems related to cyclotomic cosets are stated in Golomb. [13]

4.3 Concluding remarks

It is natural to ask whether the decomposition of the characteristic equation of an autonomous f-channel LFSR obtained by methods of Section 4.1 will give the same cycle set as the result of using those three theorems of Section 4.2. From the author's test on various examples, only one does not work exactly as it is expected. Table 4.3 summarizes all test results

of examples in this chapter. It indicates there exists at least one contradictory example. The reason is that the polynomial $\phi_5(x)$ contains repeated factors, a case pointed out by Elspas,^[10] that the circuit behavior is not uniquely determined by the characteristic polynomial. In conclusion, the cycle set of an autonomous parallel LFSR is best evaluated from the result of Section 4.1.

TABLE 4.3

$\phi_1(x)$	Cycle set (c.s.) of $\phi_1(x)$	Results of Section II Cycle set of $\phi_1(x)$ Transformed to f-channel	Result of Section III $\phi_f(x) = T^f - xI $	Does the results of II & III agree by way of Ref. 10 & 37?
$1+x+x^4$	$\{1(1), 1(15)\}$	$\{1(1), 3(5)\}^{f=3}$	$\phi_3(x) = 1+x+x^2+x^3+x^4$ c.s. = $\{1(1), 3(5)\}$	Yes
$1+x+x^4$	$\{1(1), 1(15)\}$	$\{1(1), 5(3)\}^{f=5}$	$\phi_5(x) = 1+x+x^2+x^4$ c.s. = $\{1(1), 1(3), 2(6)\}$	No *
$1+x^2+x^5$	$\{1(1), 1(31)\}$	$\{1(1), 1(31)\}^{f=2,4}$	$\phi_2(x) = \phi_4(x) = 1+x^2+x^5$ c.s. = $\{1(1), 1(31)\}$	Yes
$1+x^2+x^5$	$\{1(1), 1(31)\}$	$\{1(1), 1(31)\}^{f=3}$	$\phi_3(x) = 1+x^2+x^3+x^4+x^5$ c.s. = $\{1(1), 1(31)\}$	Yes
$1+x^2+x^5$	$\{1(1), 1(31)\}$	$\{1(1), 1(31)\}^{f=3}$	$\phi_5(x) = 1+x^2+x^3+x^5$ c.s. = $\{1(1), 1(31)\}$	Yes

* See [10]

CHAPTER 5

PARALLEL LFSR IN A F-CHANNEL SYSTEM—IMPLEMENTATION OF CYCLIC ERROR CORRECTING CODES

5.1 The use of single error correcting cyclic code in a f-channel system

In this section and the following sections, different ways of using parallel LFSR to implement cyclic error correcting codes are given. There is no claim in discovering any new codes; only the advantages of using parallel LFSR for implementating the encoder and decoder in a f-channel system are demonstrated.

In a f-channel transmission system, the key part of the encoder or decoder of a single error correcting (n, k) cyclic code is a f-channel analogy circuit. In using this, the circuit will be able to operate f times faster.

In encoding, immediately after all the k message bits have been fed into the f-channel analogy GF divider specified by $g(x)$, the contents of the registers represent the remainder which should be used as check bits. These check bits may come out of the output lines by additional shifts with all feedback circuit inhibited. However, the inhibition is not necessary as shown in a previous paper.^[22] Figure 5.1 shows the general diagram of the encoder and the codeword in a f-channel form.

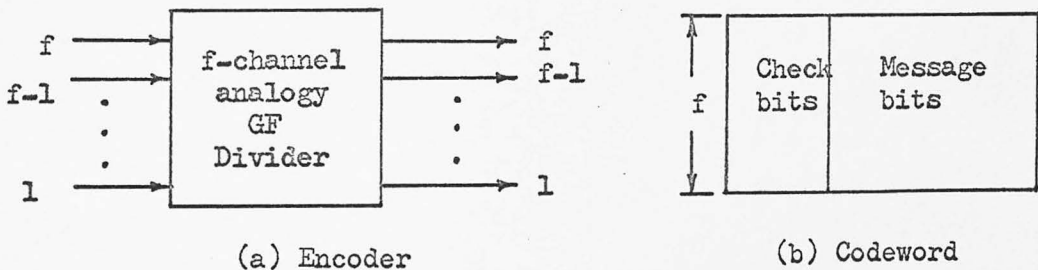


Figure 5.1 Encoder and a Codeword

The codeword is of $n = k+r$ bits long. However, when n is not divisible by f , a sufficient number of zeros (say, d) are added in the front of the codeword to make the total number of bits an integer multiple of f (say, cf). This is required in the decoding process. Let

$$cf = n + d$$

where $d \leq f-1$.

A general decoder circuit is shown in Figure 5.2. All registers are initially cleared to zero. The received $(n+d)$ bits are fed into f -channel buffer registers, each with $(n+d)/c$ stages, with SW1 closed and SW2 open. The bits are also fed simultaneously into the f -channel analogy GF divider identical to the one used in the encoder. Let H_t be the received input vector at time t , then

$$H_t = U_t + E_{ti}$$

where E_{ti} is an $1 \times f$ error vector. Since the code is a single error correcting code, E_{ti} will have a single 1 in the i -th position and 0 elsewhere. More specifically,

$$\begin{aligned} E_{t1} &= \overbrace{100 \dots 00}^f \\ E_{t2} &= 010 \dots 00 \\ &\vdots \\ E_{tf} &= 000 \dots 01 \end{aligned}$$

Otherwise, uncorrectable error pattern results.

After all $n+d$ bits are fed into the f -channel analogy circuit, the final state is given by Z ,

$$Z = H_1 G(T^f)^{c-1} + H_2 G(T^f)^{c-2} + \dots + H_{c-1} G(T^f) + H_c G \quad (5.1)$$

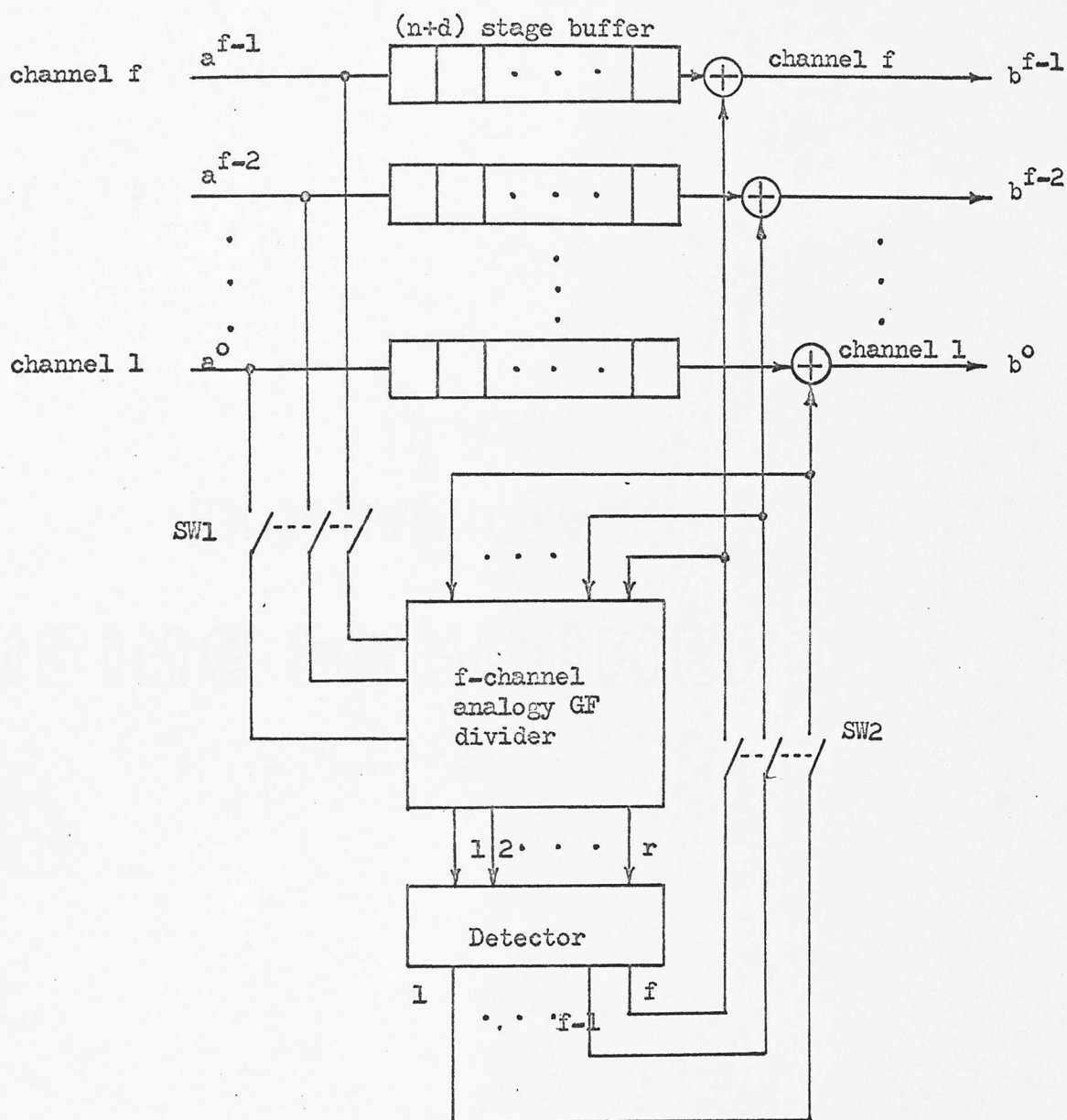


Figure 5.2 Decoder

If there has been no error, $Z = 0$, i.e.,

$$U_1 G(T^f)^{c-1} + U_2 G(T^f)^{c-2} + \dots + U_c G = 0 \quad (5.2)$$

In general, when there have been detectable errors, $Z \neq 0$. For error correction, Z should be all different for distinct errors. Let us assume that a single error in U_j ($j \in \{1, 2, \dots, c\}$) has occurred, then

$$H_j = U_j + E_{ji} \quad (5.3.a)$$

$$H_m = U_m \quad \text{for } m \neq j \quad (5.3.b)$$

Substituting Equations (5.3.a) and (5.3.b) into (5.1), we have

$$Z = E_{ji} G(T^f)^{c-j}$$

During the correction cycle, SW1 in Figure 5.2 is open and SW2 is closed. After $(j-1)$ more shifts, the erroneous group of bits H_j will arrive at the right end of the buffer register and is ready to come out at the next shift. Note that $(T^f)^{c-j} \cdot (T^f)^{j-1} = (T^f)^{c-1}$, and Z after $j-1$ shifts becomes Z_{j-1} , i.e.,

$$Z_{j-1} = E_{ji} G(T^f)^{c-1} \quad (5.4)$$

By examining the format of E_{ji} , G and $(T^f)^{c-1}$, one can easily be convinced that a single error in the i -th ($1 \leq i \leq f$) position of H_j corresponds to the i -th row of $(T^f)^{c-1}$ to form Z . Since all the rows of $(T^f)^{c-1}$ are distinct, f different AND gates A_1, A_2, \dots, A_f can be used as detectors to recognize these f different error patterns, i.e.,

$$\begin{aligned} A_1 &\approx \varphi_f (T^f)^{c-1} \\ A_2 &\approx \varphi_{f-1} (T^f)^{c-1} \\ &\vdots \\ A_i &\approx \varphi_{f-i+1} (T^f)^{c-1} \quad (i^{\text{th}} \text{ channel}) \\ A_f &\approx \varphi_1 (T^f)^{c-1} \end{aligned} \quad (5.5)$$

where the symbol \approx means "corresponding to".

During this correction cycle, if the output of the AND gate A_i is 1, then the next bit coming out of the i^{th} channel, i.e., the position b_t^{i-1} , is corrected and the f-channel analogy GF divider is reset to zero.

Example 5.1: Let us consider the Hamming (15,11) code, of which $n = 15$, $k = 11$, $r = 4$ and

$$g(x) = 1 + x + x^4, \quad f = 2$$

Solution: The 2-channel single error correction system is implemented as follows.

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The encoder circuit is essentially a 2-channel analogy GF divider which, based on the above matrices, is shown in Figure 5.3.

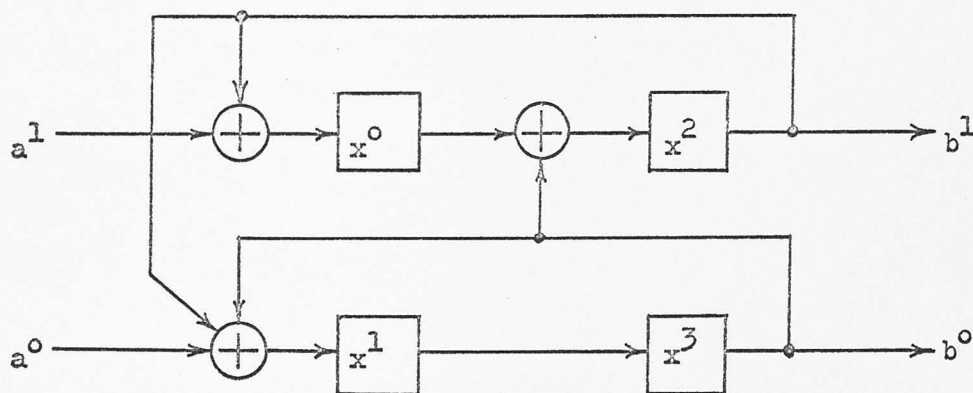


Figure 5.3 2-Channel Analogy GF Divider
 $g(x) = 1 + x + x^4$

Assume that message bits are $(11000000001) = 1 + x + x^{10}$. After this $M(x) \cdot x^4$ fed into the divider, the remainder polynomial is $r(x) = x^2 + x^3$. Therefore, the whole codeword is (001111000000001) . Since $n = 15$ is not divisible by 2, a zero ($d = 1$) is added as the leading bit to make the total number of bit a multiple of 2. Therefore, this complete 2-channel codeword is now read as follows.

$$\begin{array}{cccccccccc}
 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & t \\
 \text{Chan. 2} & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & a_t^1 \\
 \text{Chan. 1} & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & a_t^0
 \end{array}$$

That is,

$$U_1 = [01], U_2 = U_3 = U_4 = U_5 = U_8 = [00]$$

$$U_6 = U_7 = [11]$$

$$(T^2)^1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad (T^2)^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \quad (T^2)^7 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.6)$$

It can be easily verified that

$$U_1 G(T^2)^7 + U_6 G(T^2)^2 + U_7 G(T^2) = 0$$

as a consequence of the condition of a legal codeword.

The above codeword with an error added to the bit position a_4^1 , i.e., the 4th bit of channel 2, is then fed into the decoder of Figure 5.4. From Equations(5.3.a) and (5.3.b), we have

$$H_m = U_m \quad \text{for } m = 1, 2, 3, 5, 6, 7$$

$$H_4 = U_4 + E_{42}$$

where

$$E_{42} = [01]$$

This implies $Z = [01] \cdot G(T^2)^4 \neq 0$

Note that the two detector AND gates A_1 and A_2 are corresponding to the two rows of the matrix $(T^2)^7$, i.e.,

$$A_1 \sim 1000, \quad A_2 \sim 1001$$

Since $Z \neq 0$ at the end of detection cycle, the correction cycle starts. After three more shifts, the erroneous bit a_t^1 reaches the right-most bit position of the channel 2, and at this time, from Equation (5.4) which is exactly recognized by the AND gate A_2 and the

$$Z = E_{42} G(T^2)^7$$

error is corrected when it is shifted out during the next pulse time.

Table 5.1 shows how the error correction is performed.

Table 5.1 Error Correction State Table

shift pulse		inputs		state	output	
t		a_t^0	a_t^1	$x_t^0 x_t^1 x_t^2 x_t^3$	b_t^0	b_t^1
Detection	0	0	0	0 0 0 0		
cycle	1	0	1	1 0 0 0		
(SW1 closed)	2	0	0	0 0 1 0		
(SW2 open)	3	0	0	1 1 0 0		
	4	0	1*	1 0 1 1		
	5	0	0	1 0 0 0		
	6	1	1	1 1 1 0		
	7	1	1	0 0 1 1		
	8	0	0	1 0 1 0		
Correction	0			1 0 1 0	0	0
cycle	1			1 1 1 0	0	1
(SW1 open)	2			1 1 1 1	0	0
(SW2 closed)	3			1 0 0 1 (emit 1)	0	0
	4			0 0 0 0	0	0 **
	5			0 0 0 0	0	0
	6			0 0 0 0	1	1
	7			0 0 0 0	1	1
	8			0 0 0 0	0	0

*bit in error

**error correction

5.2 A straightforward 2-channel BCH (15,7) decoder

In this section, a straightforward way of constructing a decoder of the Bose-Chaudhuri-Hocquenham (15,7) double error correcting code is given. It is different from the algorithm given by Peterson, [3] improved by Chien [39] and many others. The system given here is merely to show another possibility of using parallel LFSR as a central device to correct double random errors in a f-channel parallel transmission system. Moreover, as different from the previous section, in this section the f-channel analogy SRSG is used.

The characteristic polynomial for a t-error-correcting BCH code is

$$\phi(\lambda) = \phi_1(\lambda) \phi_3(\lambda) \phi_5(\lambda) \dots \phi_{2t-1}(\lambda)$$

Therefore, the two error correcting BCH (15,7) codes have the following characteristic polynomial:

$$\phi(\lambda) = \phi_1(\lambda) \phi_3(\lambda)$$

where $\phi_1(\lambda)$ is always chosen to be primitive and its degree m has to satisfy the condition $n=2^m-1$. Therefore, by letting

$$\phi_1(\lambda) = \lambda^4 + \lambda + 1$$

and using the result of Chapter 4, we have

$$\begin{aligned} \phi_3(\lambda) &= \lambda^4 + \lambda^3 + \lambda^2 + \lambda + 1 \\ \therefore \phi(\lambda) &= \lambda^8 + \lambda^7 + \lambda^6 + \lambda^4 + 1 \end{aligned} \tag{5.7}$$

The characteristic polynomial of Equation (5.7) is actually the generator polynomial $g(x)$ of the code. The encoder is simply a 2-channel analogy SRSG as shown in Figure 5.5 derived from the following set of matrices.

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Therefore,

$$\begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \\ x^7 \end{bmatrix}^{t+1} = \begin{bmatrix} x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \\ x^7 \\ x^0 + x^4 + x^6 + x^7 + a^0 \\ x^0 + x^1 + x^4 + x^5 + x^6 + a^0 + a^1 \end{bmatrix}^t$$

and the circuit is then shown below.

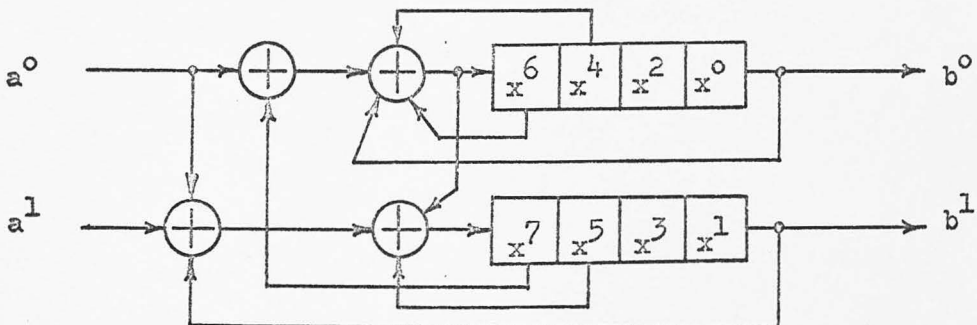


Figure 5.5 The 2-Channel Analog SRSG
 $g(x) = x^8 + x^7 + x^6 + x^4 + 1$

Next, the correction cycle starts by opening the SW1 and closing SW2. After (i-1) more shifts, the H'_i moves into the position of H'_1 , i.e., the right-most position of the buffer register. Therefore,

$$Z'_{i-1} = (T^2)^7 L E'_{iq} + (T^2)^{7-Q} L E'_{jq} \tag{5.10}$$

where $Q \in \{0, 1, 2, \dots, 7\}$: Equation (5.10) includes all possible double error patterns. The detector circuit is then constructed on the basis of this equation. However, since the number of patterns to be recognized is 27, Table 5.3 is given to list all possible patterns.

First, the table for all patterns of the term $(T^2)^{7-Q} L E'_{jq}$ is constructed as follows.

Table 5.2 Format of $(T^2)^{7-Q} L E'_{jq}$

Q	pattern #	$(T^2)^{7-Q} L E'_{j1}$	pattern #	$(T^2)^{7-Q} L E'_{j2}$
		$\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ x & x & x & x & x & x & x & x & x \end{matrix}$		$\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ x & x & x & x & x & x & x & x & x \end{matrix}$
0	1	0 0 0 0 0 0 0 1	9	1 0 0 0 0 0 0 0
1	2	0 1 0 0 0 0 0 0	10	0 0 1 0 0 0 0 0
2	3	0 0 0 1 0 0 0 0	11	1 0 0 0 1 0 0 0
3	4	0 1 0 0 0 1 0 0	12	1 0 1 0 0 0 1 0
4	5	1 1 0 1 0 0 0 1	13	0 1 1 0 1 0 0 0
5	6	0 0 1 1 0 1 0 0	14	0 0 0 1 1 0 1 0
6	7	0 0 0 0 1 1 0 1	15	0 0 0 0 0 1 1 0
7	8	0 0 0 0 0 0 1 1	16	0 0 0 0 0 0 0 1

Note that $E'_{j1} = [10]'$

$E'_{j2} = [01]'$

From Table 5.2, all possible double error patterns can be derived. The total number of possible double error patterns is equal to $1 + 13 + 13 = 27$. All these patterns are listed in Table 5.3. Note that there are two sets of detecting circuits; the first set is to correct the first error bit in the upper channel and the second set is to correct the first error bit in the lower channel.

Table 5.3 Double Error Pattern

pattern #		$\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ x & x & x & x & x & x & x & x \end{matrix}$
1	1+9	1 0 0 0 0 0 0 1
2	1+2	0 1 0 0 0 0 0 1
3	1+3	0 0 0 1 0 0 0 1
4	1+4	0 1 0 0 0 1 0 1
5	1+5	1 1 0 1 0 0 0 0
6	1+6	0 0 1 1 0 1 0 1
7	1+7	0 0 0 0 1 1 0 0
8	1+8	0 0 0 0 0 0 1 0
9	1+10	0 0 1 0 0 0 0 1
10	1+11	1 0 0 0 1 0 0 1
11	1+12	1 0 1 0 0 0 1 1
12	1+13	0 1 1 0 1 0 0 1
13	1+14	0 0 0 1 1 0 1 1
14	1+15	0 0 0 0 0 1 1 1
15	9+2	1 1 0 0 0 0 0 0
16	9+3	1 0 0 1 0 0 0 0
17	9+4	1 1 0 0 0 1 0 0
18	9+5	0 1 0 1 0 0 0 1
19	9+6	1 0 1 1 0 1 0 0
20	9+7	1 0 0 0 1 1 0 1
21	9+8	1 0 0 0 0 0 1 1
22	9+10	1 0 1 0 0 0 0 0
23	9+11	0 0 0 0 1 0 0 0
24	9+12	0 0 1 0 0 0 1 0
25	9+13	1 1 1 0 1 0 0 0
26	9+14	1 0 0 1 1 0 1 0
27	9+15	1 0 0 0 0 1 1 0

In Table 5.3, the pattern 1 is common to both upper and lower channels, while the patterns 2, 3, ..., 14 are for the first error bit in the upper channel and the patterns 15, 16, ..., 27 are for the first error bit in the lower channel. Let the AND gates A_1, A_2, \dots, A_{27} having 8 inputs corresponding to each pattern, then A_1, A_2, \dots, A_{14} is "ORed" together to become the upper channel detector and similarly for the lower channel detector. These two detectors may have some further logical simplification because there are multiple input-output combinational network and standard techniques of simplification are available. [40][41]

After the first error bit of the double errors has been corrected, then the 2-channel analogy SRSQ make (j-1) more shifts which forces the second error bit reaching the right-most bit position. This error is then corrected as the single error case. Note that the single error detector is to recognize the following two patterns:

$$00000001, 10000000$$

This completes our theoretical design work.

5.3 Burst error correction in a f-channel transmission system

The principles used in previous sections can be applied to correct a burst of b errors. Let a positive integer m be the number of time periods during which the burst of errors occurs ($mf \geq b$). Let the m error vectors be W_1, W_2, \dots, W_m with W_i leading W_{i+1} accordingly. Note here that any W-vector may have more than a single one as its entry and hence is a general form of the single error vector E_{ti} . Again, only the decoding procedures are discussed below.

After the detection cycle, a detectable error pattern $Z \neq 0$ occurs which starts the correction cycle. After (j-1) more shifts, the state of the f-channel analogy GF divider becomes

$$Z_{j-1} = W_1 G(T^f)^{-1} + W_2 G(T^f)^{-2} + \dots + W_m G(T^f)^{-m} \quad (5.11)$$

Equation (5.11) can be easily derived from Equations (5.2) and (5.3) by noting $T^n = I$. Equation (5.11) is corresponding to the sum modulo-2 of some of the first f rows of the matrices $(T^f)^{-1}, (T^f)^{-2}, \dots, (T^f)^{-m}$. In general, there are $f \cdot 2^{b-1}$ error patterns to be detected although some simplification is usually possible.

Upon recognition of a detectable error pattern, bits are emitted into the appropriate channels to correct the first error vector W_1 , which is just

ready to come out the buffer registers. This effectively modifies the state of the GF divider Z_{j-1} into

$$(Z_{j-1} + W_1 G(T^f)^{-1}) (T^f) = W_2 G(T^f)^{-1} + W_3 G(T^f)^{-2} \\ + \dots + W_m G(T^f)^{-m+1}$$

after one feedback shift. This again belongs to one of the $f \cdot 2^{b-1}$ detectable patterns. The process continues until all the errors are corrected, at which time the registers of the divider should contain all zeros. Otherwise the presence of some uncorrectable errors is indicated.

Note that Equation (5.11) indicates that all the error patterns represented by the equation are distinct and none of them appears in any of the previous shifts. This requirement is fulfilled because they are precisely the conditions for a proper choice of the code.

Example 5.2: Given $g(x) = (1 + x^3)(1 + x^3 + x^4) = 1 + x^4 + x^6 + x^7$
and $f = 3$

then

$$T = \begin{bmatrix} 0100000 \\ 0010000 \\ 0001000 \\ 0000100 \\ 0000010 \\ 0000001 \\ 1000101 \end{bmatrix}, \quad T^3 = \begin{bmatrix} 0001000 \\ 0000100 \\ 0000010 \\ 0000001 \\ 1000101 \\ 1100111 \\ 1110110 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 0010000 \\ 0100000 \\ 1000000 \end{bmatrix}, \quad J = \begin{bmatrix} 000 \\ 000 \\ 000 \\ 001 \\ 011 \\ 111 \end{bmatrix}$$

The 3-channel analogy GF divider specified by the above matrices is shown in Figure 5.6.

Assume that the message polynomial is $x^6 + x^4 + x^2 + 1$ (01010101), the complete codeword becomes 010101011011111, i.e.,

$$U_1 = [010], U_2 = [101], U_3 = [011], U_4 [011], U_5 = [111]$$

Suppose that the received message is 010 111 111 011 111 with two errors

a_2^1 and a_3^0 . Then,

$$W_2 = [010], \quad W_3 = [100]$$

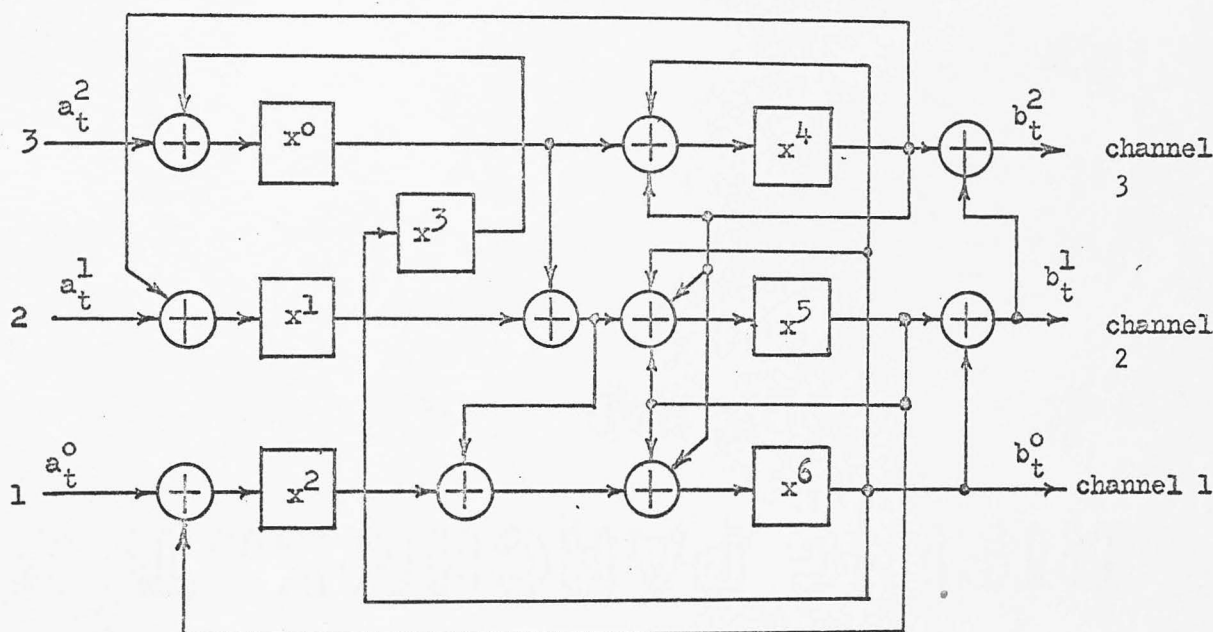


Figure 5.6 3-Channel Analogy GF Divider

$$g(x) = 1 + x^4 + x^6 + x^7$$

$$(T^f)^{-1} = \begin{bmatrix} 0101100 \\ 0010110 \\ 0001011 \\ 1000000 \\ 0100000 \\ 0010000 \\ 0001000 \end{bmatrix}, \quad (T^f)^{-2} = \begin{bmatrix} 1110110 \\ 0111011 \\ 1011000 \\ 0101100 \\ 0010110 \\ 0001011 \\ 1000000 \end{bmatrix}$$

The state transition table of the decoding process is shown in Table 5.4. The content of the divider is not all zero at the end of the detection cycle. After one more feedback shift, the state becomes 1001110,

which is one of the error patterns to be detected. It can be verified that

$$W_2 G(T^f)^{-1} + W_3 G(T^f)^{-2} = 0010110 + 1011000 \\ = 1001110$$

At this time, the W_2 associated error vector reaches the right-most position of the buffer register and is ready to shift out. The detector emits a 1 into the channel 2 to correct the error bit a_2^1 and also into the input a_t^1 . After one feedback shift, the state of the divider is 0001011, which is also one of the detectable error patterns and is the third row of $(T^f)^{-1}$. This time the detector emits a 1 into the channel 1 to correct the bit a_3^1 and also to clear the contents of the divider to all zero for next coming codeword.

Table 5.4 Decoding Process Table

	shift pulse t	input $\begin{smallmatrix} 0 & 1 & 2 \\ a_t & a_t & a_t \end{smallmatrix}$	state of divider $\begin{smallmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ x & x & x & x & x & x & x \end{smallmatrix}$	pulse emitted by detector	output $\begin{smallmatrix} 0 & 1 & 2 \\ b_t & b_t & b_t \end{smallmatrix}$
Detection	0		0 0 0 0 0 0 0		
cycle	1	0 1 0	0 1 0 0 0 0 0		
SW1	2	1 1* 0	1 1 1 0 1 0 0		
closed,	3	1* 1 1	0 1 1 1 0 1 1		
SW2	4	0 1 1	1 1 1 0 1 1 0		
open	5	1 1 1	1 0 1 1 1 0 0		
Correction	6		1 0 0 1 1 1 0		0 1 0
cycle	7		0 0 0 1 0 1 1	0 1 0	1 0 1
SW1	8		0 0 0 0 0 0 0	1 0 0	0 1 1
open,	9				0 1 1
SW2	10				1 1 1
closed					

* bit in error

CHAPTER 6

SINGLE CHANNEL ERROR CORRECTION IN A F-CHANNEL SYSTEM

6.1 General system description

In this chapter, a method of correcting errors occurring in a single channel of a f-channel system is disclosed. There is no restriction either on the length or type (random or burst) of the error pattern. In a practical case, this one channel error is defined within a block. The following Figure 6.1 shows the general picture of blocks.

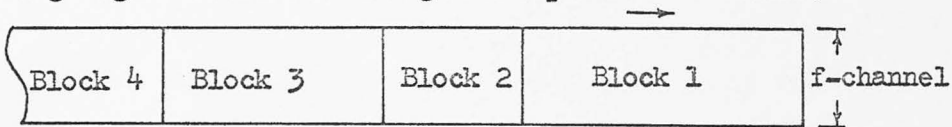


Figure 6.1

Note that the blocks do not have to be equal length. In each block, the only requirement is that errors only occur in a single channel among the f-channels. Strictly speaking, there may be uncorrectable single channel error patterns, but the percentage of these uncorrectable patterns can be made arbitrarily small. This type of error correction utilizing the principles of parallel LFSR can be easily applied to data processing systems such as tape, disk and drum. It can be applied also to the multi-channel communication system.

The assumption that error occurs only in one channel within a block is practically justified. In data processing systems, such as high density recorded tapes or disks, the space between channels is much greater than the bits crowded in one channel. For example, in the IBM 360/2400 tape system, the space between the adjacent channel centers is about 40 times the space between the adjacent bits in a channel. Based on this assumption, the error correcting system is formed in the following way.

1. Among the f -channels, one channel denoted by p -channel, must be an even or odd parity channel.
2. A cyclic code is formed with the overall $f-1$ channels* but is not associated with each channel separately.

The general transmitting system is shown in Figure 6.2.

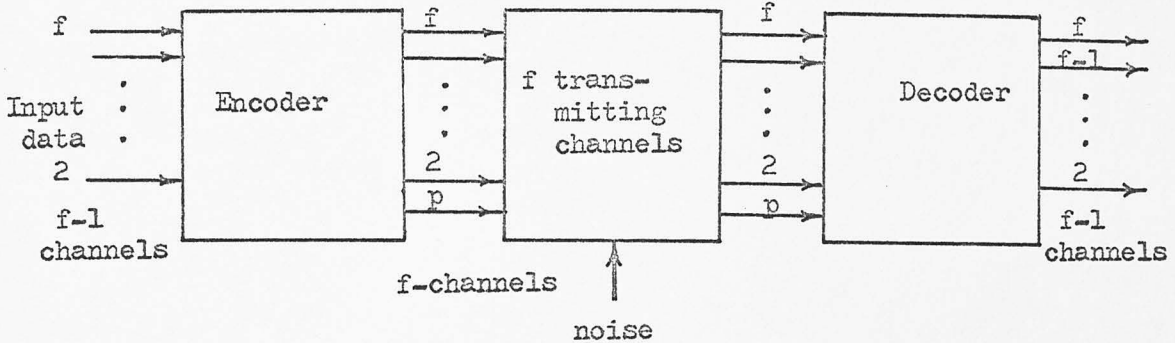


Figure 6.2 General System Diagram

The block diagrams for the encoder and the decoder are shown in Figures 6.3 and 6.4, respectively.

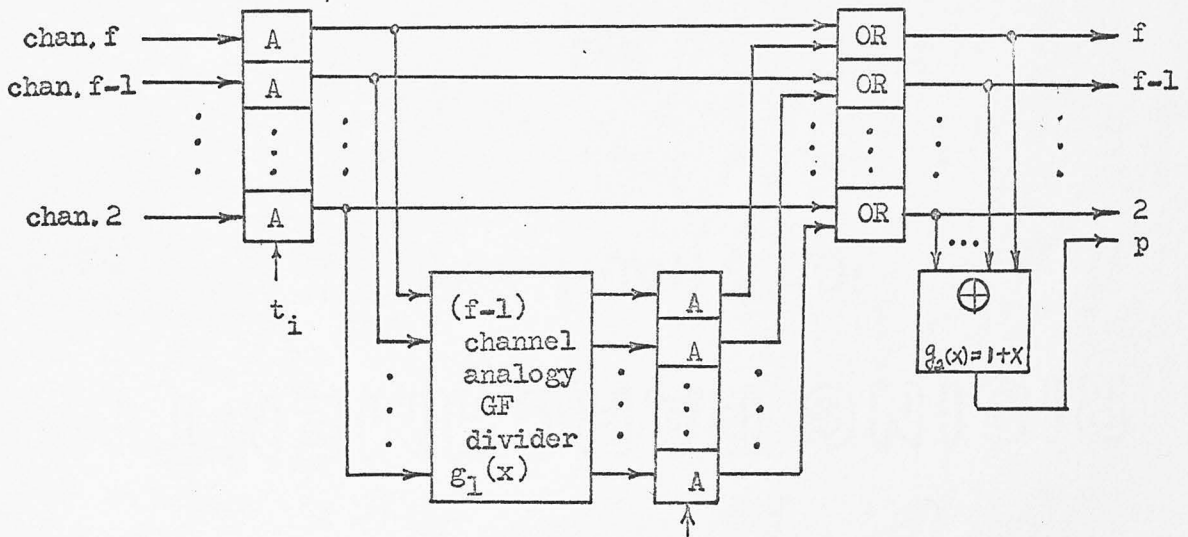


Figure 6.3 Encoder

* The p -channel may be included

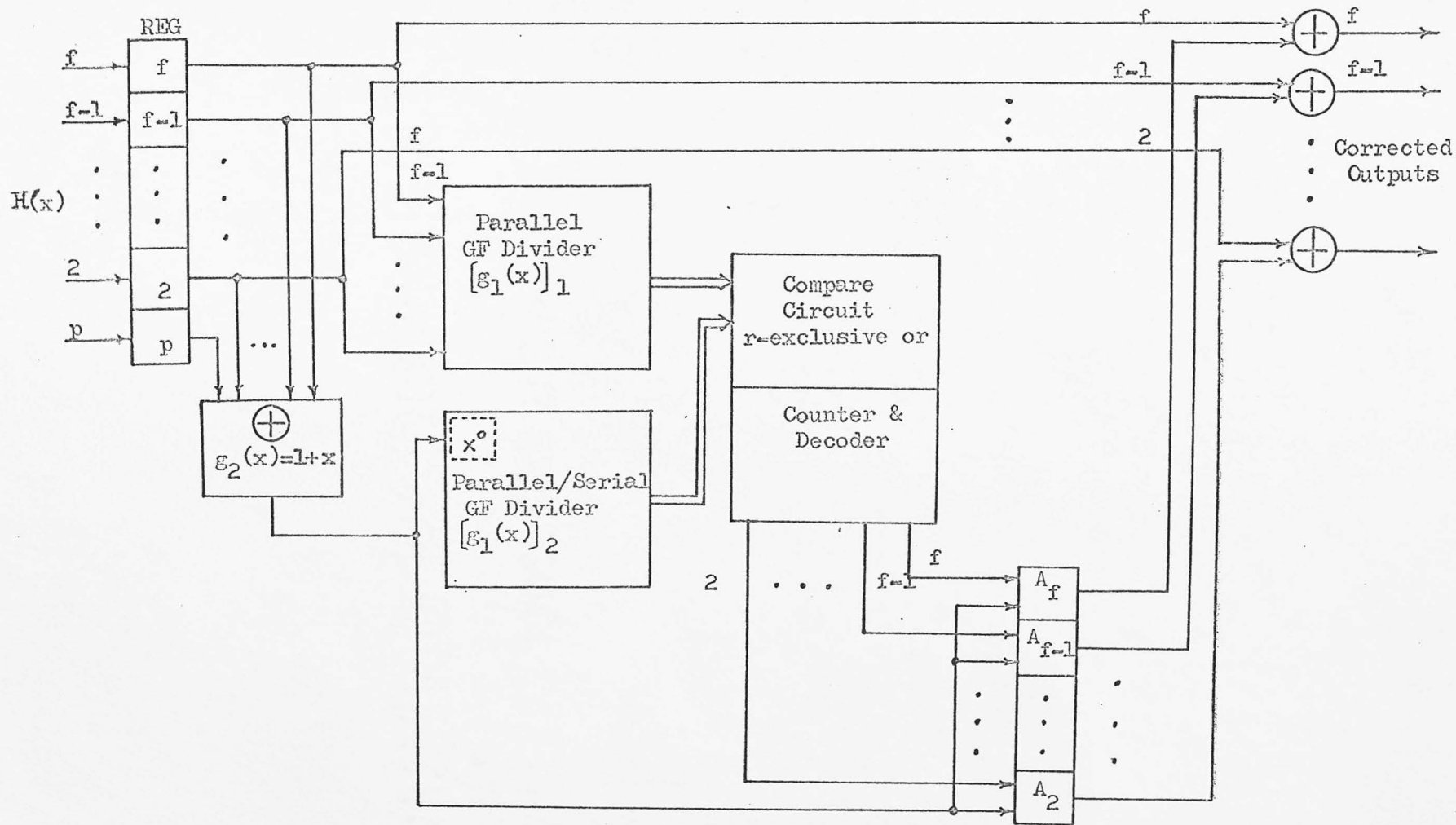


Figure 6.4 Decoder

6.2 The 2-dimensional variable length code (2-DVLC)

The symbology and terminology and definitions on coding theory used in this chapter are essentially following Peterson. [3]

Let

$v(x)$: message polynomial of degree $k-1$, $v(x) = a^0 x^{k-1} + a^1 x^{k-2} + \dots + a^{k-2} x + a^{k-1}$

$F(x)$: codeword polynomial of degree $n-1$, $F(x) = a^0 x^{n-1} + a^1 x^{n-2} + \dots + a^{n-2} x + a^{n-1}$

$H(x)$: received polynomial, $H(x) = F(x) + e(x)$

$e(x)$: error polynomial, $e(x) = x^\ell (1 + \dots) = x^\ell E(x)$, ℓ is a positive integer.

$E(x)$: error pattern polynomial, $E(x) = 1 + \dots$

$e'(x)$: shifted error polynomial, $e'(x) = x^{\ell-j} E(x)$ where j is an integer and $0 \leq j \leq f-1$

The code given here is of variable length and of 2-dimension nature.

Its general polynomial $g(x)$ is defined below

$$g(x) = g_1(x) * g_2(x) \quad (6.1)$$

where the "*" sign is a composite operation formulated as shown in Figure 6.5.

a^{n-1}	a^{k-1}	a^{f-2}	f
Remainder $r(x)$ due to $V(x)/g_1(x)$	$V(x)$	a^f	a^1
		a^{f-1}	a^0
Parity check bit on columns \equiv Remainder due to $g_2(x)$ on columns			p

Figure 6.5 A Codeword Block

In Figure 6.5, there are two kinds of checkbits, the first kind is the remainder $r(x)$ due to $v(x)$ after divided by $g_1(x)$, while the second kind is

due to $g_2(x)$ which acts column-wise on both $v(x)$ and the first kind of check bits. Here $g_1(x)$ of degree r is the polynomial that determines the undetected error fraction $2^{-(n-k)}$. It is also used to identify the erroneous channel. Normally, $g_1(x)$ is chosen from a primitive polynomial or a primitive reciprocal polynomial. Note that the check bits formed by the remainder of $g_1(x)$ on the entire message block is different from the conventional way. In the conventional way, the check bits are formed with respect to each channel. Moreover, the Hamming distance of the code given here is not the product of the distances due to $g_1(x)$ and $g_2(x)$. Actually, the error correction capability of the system is not due to the code itself alone; it is greatly helped from the operation of the LFSR's.

The error correction process requires the cooperation of $g_1(x)$ and the $g_2(x)$. Here $g_2(x) = 1 + x$.

This error correction process can be divided into three cycles.

1. The detection cycle
2. The comparison cycle
3. The correction cycle

The flow chart diagram of the operation of these three cycles is shown in Figure 6.6. In the conventional error correction process utilizing cyclic codes, the comparison cycle does not exist. It is the addition of this comparison cycle that makes the unlimited error correction on one channel of the system possible.

Since the data blocks are of variable length, the polynomial $v(x)$ of length k is used to represent a typical block for discussion. The encoder shown in Figure 6.3 contains a $f-1$ channel analogy GF divider specified by $g_1(x)$. The remainder $r(x)$ thus produced is attached to the end of $v(x)$. The remainders of $g_2(x)$ are obtained column-wise from the channels 2 to f

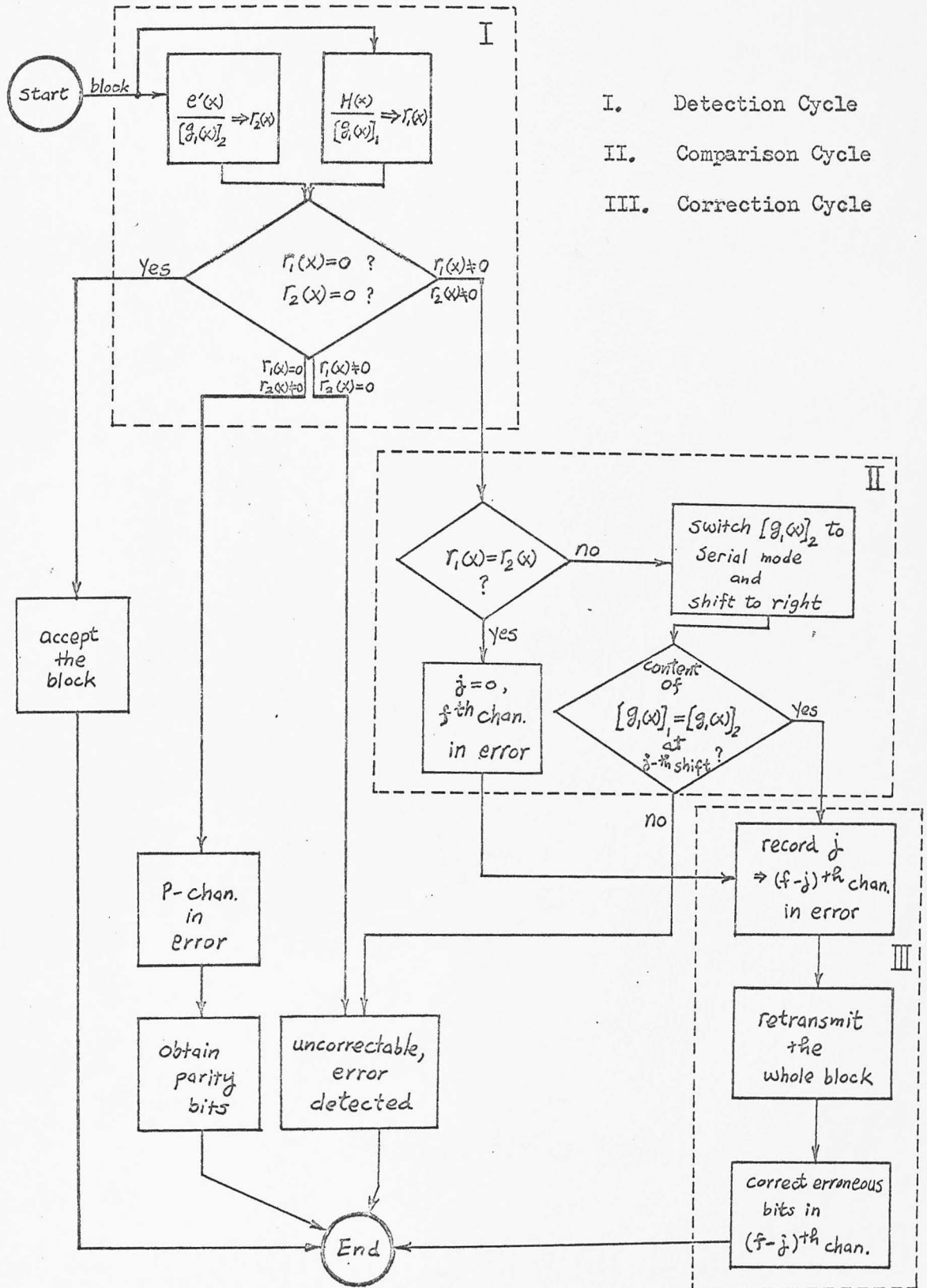


Figure 6.6 Decoding Procedures in Flow-Diagram Form

to form the p-channel or the parity channel.

The input polynomial $v(x)$ with $f-1$ bits coming in each clock pulse time passes through the encoder to form a polynomial $F(x)$ and the p-channel which constitute the whole block codeword. Here,

$$F(x) = x^f v(x) + r(x)$$

or,
$$\frac{x^f v(x)}{g_1(x)} = q(x) + \frac{r(x)}{g_1(x)}$$

The f -channel codeword block is then transmitted through f channels and changed by noises. Since the basic assumption is that errors affect only on one channel (not the p-channel) in the block,* the received block becomes $H(x)$ and

$$H(x) = F(x) + e(x)$$

Figure 6.7 shows an $e(x)$ and its shifted $e'(x)$.

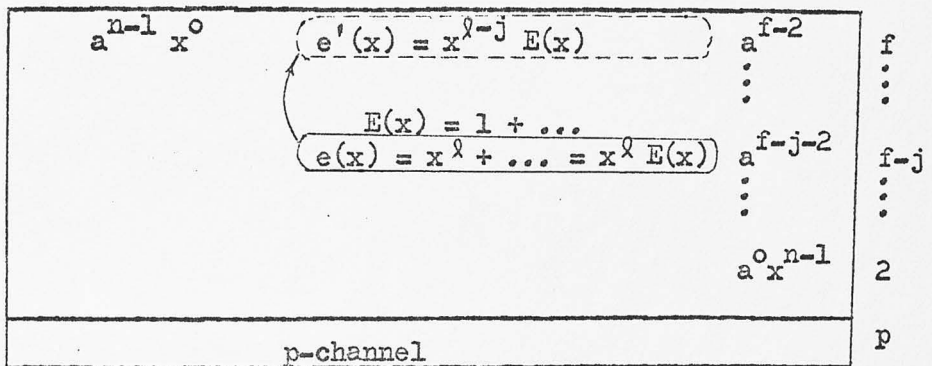


Figure 6.7 $e(x)$ and $e'(x)$ in a Codeword Block

The received codeword with the noise is now fed into the decoder. In Figure 6.4, the decoder contains two GF dividers. The first one is a $f-1$ channel analogy circuit denoted by $[g_1(x)]_1$. The second one operates in two modes. Its first mode operating in the detection cycle is the same as

* If errors are occurred in the p-channel, see Figure 6.6 for a solution.

the $[g_1(x)]_1$ except with only one input line to the x^0 -position, i.e.,

$$G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \zeta_1(I) \end{bmatrix}$$

instead of using Equation (2.16.a). Here x^0 -position is selected as a reference channel to identify the erroneous channel. It is only a matter of convenience for the proof in Section 6.3. Its second mode operating in the comparison cycle is a serial GF divider. Since both modes are characterized by $g_1(x)$, the second GF divider is denoted by $[g_1(x)]_2$ and its first mode is called the parallel mode and the second mode is called the serial mode.

The content of $[g_1(x)]_1$, after $H(x)$ entering the circuit during the detection cycle, is solely determined by the error polynomial $e(x)$. This is true since the system is linear and therefore $F(x)$ and $e(x)$ can be treated separately. While at the same time, the content of the parallel mode of $[g_1(x)]_2$ is determined by the shifted error polynomial $e'(x)$ fed into the x^0 -position. From Figure 6.7, we have

$$e'(x) = x^{k-j} E(x) \quad (6.2)$$

Therefore the difference between the $e(x)$ and $e'(x)$ is a factor x^j .

At the end of detection cycle, if no error occurs, then the contents of the $[g_1(x)]_1$ and $[g_1(x)]_2$ should be all zero after $H(x)$ and p -channel bits come in. If errors have occurred among one of the data channel, then $e(x) \neq 0 \Rightarrow e'(x) \neq 0$ and the content of $[g_1(x)]_1$ and $[g_1(x)]_2$ will not be all zero.

When the nonzero patterns are detected at the end of the detection cycle, the decoder begins its comparison cycle, i.e.,

1. No input to $[g_1(x)]_1$ and $[g_1(x)]_2$.
2. The content of the $[g_1(x)]_1$ is fixed.
3. The circuit of $[g_1(x)]_2$ is switched into its serial mode and begins to shift right. This feedback shifting process will not be stopped until the content of $[g_1(x)]_2$ matches that of the $[g_1(x)]_1$.

The number of shifts of $[g_1(x)]_2$ is used to determine which channel is in error. If the number of shifts of $[g_1(x)]_2$ is j ($j < f$), then the $(f-j)$ th channel is in error. A rigorous proof on the above statement is in the next section. However, it can be argued in the following way.

Since to shift the content of a serial GF divider $g_1(x)$ one time to the right is equivalent to multiply in the present content of the LFSR by $x \bmod g_1(x)$. Therefore to shift j -times is equivalent to multiply in the present content by $x^j \bmod g_1(x)$ which is exactly the residue difference between $[g_1(x)]_1$ and $[g_1(x)]_2$.

Whence the erroneous channel is identified, the decoder begins its correction cycle, i.e.,

1. The whole received block is retransmitted.
2. The error polynomial $e(x)$ is in the $(f-j)$ th channel which is corrected by the output bits of the AND gate A_{f-j} .

During the block retransmission period, an assumption is made that errors will remain in the same channel and its pattern may be changed. With this assumption, the above mentioned error correction process is simply performed by changing the bit in the $(f-j)$ th channel whenever the parity bit of that column is wrong. In a tape or other data recording system, errors are permanent that it will not be changed during the retransmission period. Furthermore, the retransmission operation is easily achieved by winding the tape backward and then forward.

In summary, the error correction is accomplished as follows.

1. The single channel error is detected at the end of the detection cycle.
2. The erroneous channel is identified during the comparison cycle.
3. Errors are corrected in the correction cycle by retransmitting the whole block with the knowledge of which channel is in error; and the error pattern calculated from the mod 2 adder which is the circuit of $g_2(x)$ in the decoder and its inputs are the received data of all f-channels.

6.3 Mathematical justification

In this section, we shall show that if an error pattern polynomial $E(x)$ occurs in the $(f-j)$ th channel, it can be identified by the comparison cycle discussed in the previous section.

Theorem 6.1 If two input polynomials $e(x)$ and $x^{-j} e(x)$ are divided by another polynomial $g_1(x)$ to obtain two remainder polynomials $r_1(x)$ and $r_2(x)$ respectively, then by shifting the content of the $[g_1(x)]_2$ j times, we have

$$r_1(x) \equiv r_2(x) \cdot x^j \pmod{g_1(x)} \quad (6.3)$$

Proof: Suppose the lowest term of the error polynomial $e(x)$ is x^l , then

$$e(x) = x^l E(x)$$

and from Equation (6.2),

$$e'(x) = x^{l-j} E(x) = x^{-j} e(x) \quad (6.4)$$

Referring to Figure 6.4, the content of $[g_1(x)]_1$, after $H(x)$ fed in, is equal to $r_1(x)$, where

$$\begin{aligned} \frac{H(x)}{g_1(x)} &= \frac{F(x) + e(x)}{g_1(x)} \equiv \frac{e(x)}{g_1(x)} \pmod{g_1(x)} \\ \frac{e(x)}{g_1(x)} &= q_1(x) + \frac{r_1(x)}{g_1(x)} \end{aligned} \quad (6.5)$$

Next, consider that $e'(x)$ is fed into the x^0 -position of $[g_1(x)]_2$, we have

$$\frac{e'(x)}{g_1(x)} = q_2(x) + \frac{r_2(x)}{g_1(x)}$$

or,

$$\frac{x^{-j} e(x)}{g_1(x)} = q_2(x) + \frac{r_2(x)}{g_1(x)} \quad (6.6)$$

To shift $r_2(x)$ in the $[g_1(x)]_2$ for j times is equivalent to multiplying $r_2(x)$ by $x^j \bmod g_1(x)$. Therefore, from Equation (6.6), we have

$$x^j \cdot \left(x^{-j} \frac{e(x)}{g_1(x)} \right) = x^j q_2(x) + x^j \frac{r_2(x)}{g_1(x)}$$

or,

$$\frac{e(x)}{g_1(x)} = x^j q_2(x) + \frac{x^j r_2(x)}{g_1(x)} \quad (6.7)$$

Comparing Equations (6.5) and (6.7), we have

$$q_1(x) \equiv q_2(x) \cdot x^j \bmod g_1(x)$$

and

$$r_1(x) \equiv r_2(x) \cdot x^j \bmod g_1(x) \quad (6.8)$$

Equation (6.8) is the desired result. Therefore, the content of $[g_1(x)]_2$ will match that of the $[g_1(x)]_1$ if the number of times of feedback shifts of the serial mode of $[g_1(x)]_2$ is exactly j times.

In the decoder, the number j is recorded by the counter. During the correction cycle, this decoded j and the error pattern coming out of the mod 2 adder sense the AND gate A_{f-j} to correct all errors in the $(f-j)$ th channel.

6.4 An example for illustration

One special example will be helpful to clarify most previous statements. Let us choose a block of data, such as, 1011,0101, 0010,0001,1101 for a 4-channel parallel transmission system. The additional p-channel makes $f=5$, then

$$v(x) = x^{19} + x^{17} + x^{16} + x^{15} + x^{10} + x^7 + x^5 + x^3 + x^2 + 1$$

and let

$$g_1(x) = x^8 + x^5 + x^3 + x + 1$$

then

$$\frac{x^8 v(x)}{g_1(x)} = q(x) + \frac{x^6 + x^4 + x^3 + x}{g_1(x)}$$

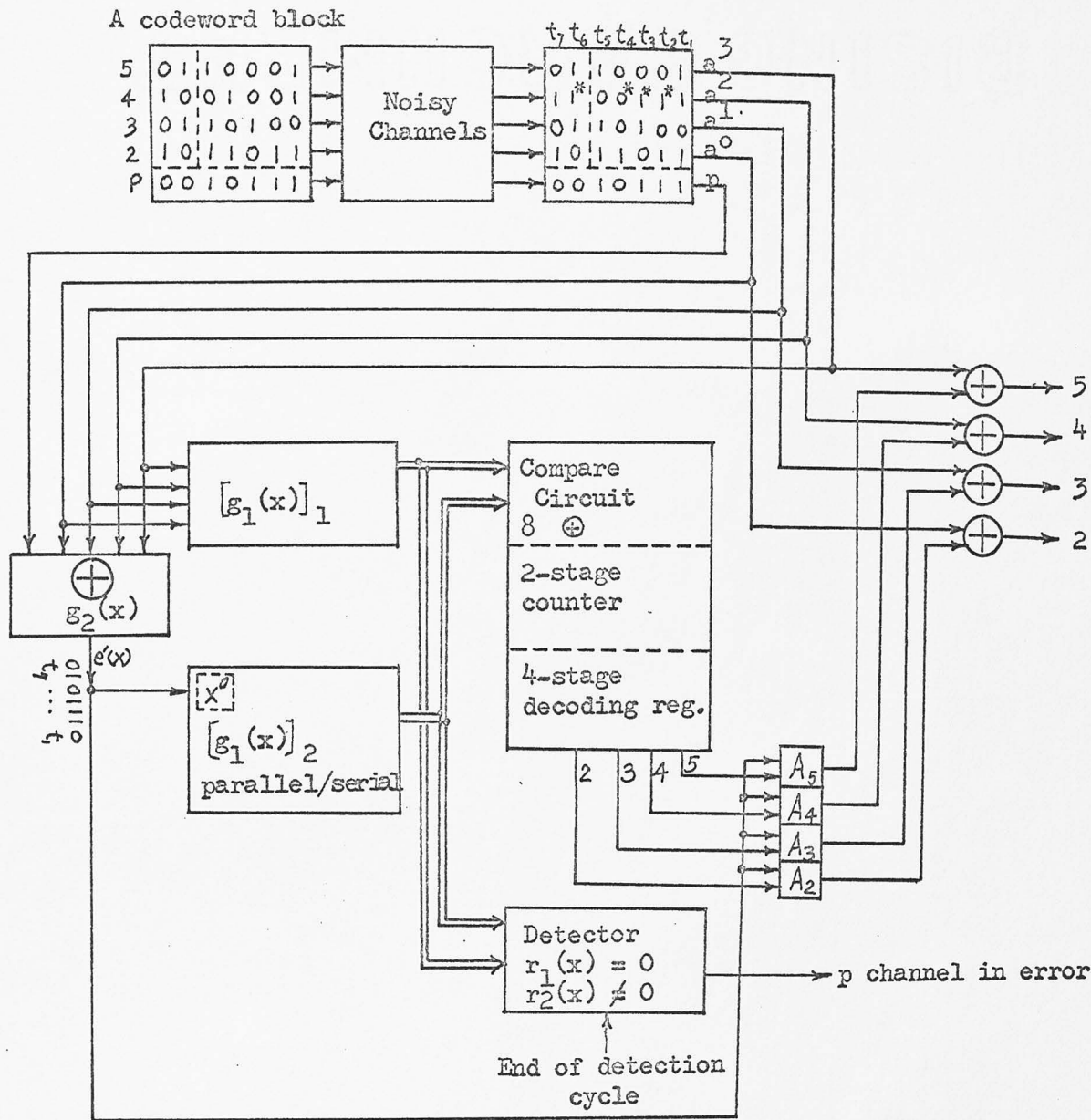
Therefore,

$$F(x) = x^{27} + x^{25} + x^{24} + x^{23} + x^{18} + x^{15} + x^{13} \\ + x^{11} + x^{10} + x^8 + x^6 + x^4 + x^3 + x$$

The encoder circuit contains a mod 2 adder and a 4-channel analogy GF divider specified by $g_1(x)$ and is omitted here. However, the 4-channel analogy circuit is also required by the decoder except the output of the circuit is not needed in the decoder.

Figure 6.8 is the decoder diagram with the assumed codeword block and the error polynomial $e(x)$ as indicated.

$$F(x) = x^{27} + x^{25} + x^{24} + x^{23} + x^{18} + x^{15} + x^{13} \\ + x^{11} + x^{10} + x^8 + x^6 + x^4 + x^3 + x \\ e(x) = x^{21} + x^{17} + x^{13} + x^5$$



* bit in error

Figure 6.8 Decoder

In Figure 6.8, the 4-channel analogy GF divider is constructed as follows.

Since $g_1(x) = x^8 + x^5 + x^3 + x + 1$

$$T = \begin{bmatrix} 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 11010100 \end{bmatrix}, \begin{bmatrix} 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 11010100 \\ 01101010 \\ 00110101 \\ 11001110 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 00010000 \\ 00100000 \\ 01000000 \\ 10000000 \end{bmatrix}$$

Therefore, the circuit is shown in Figure 6.9.

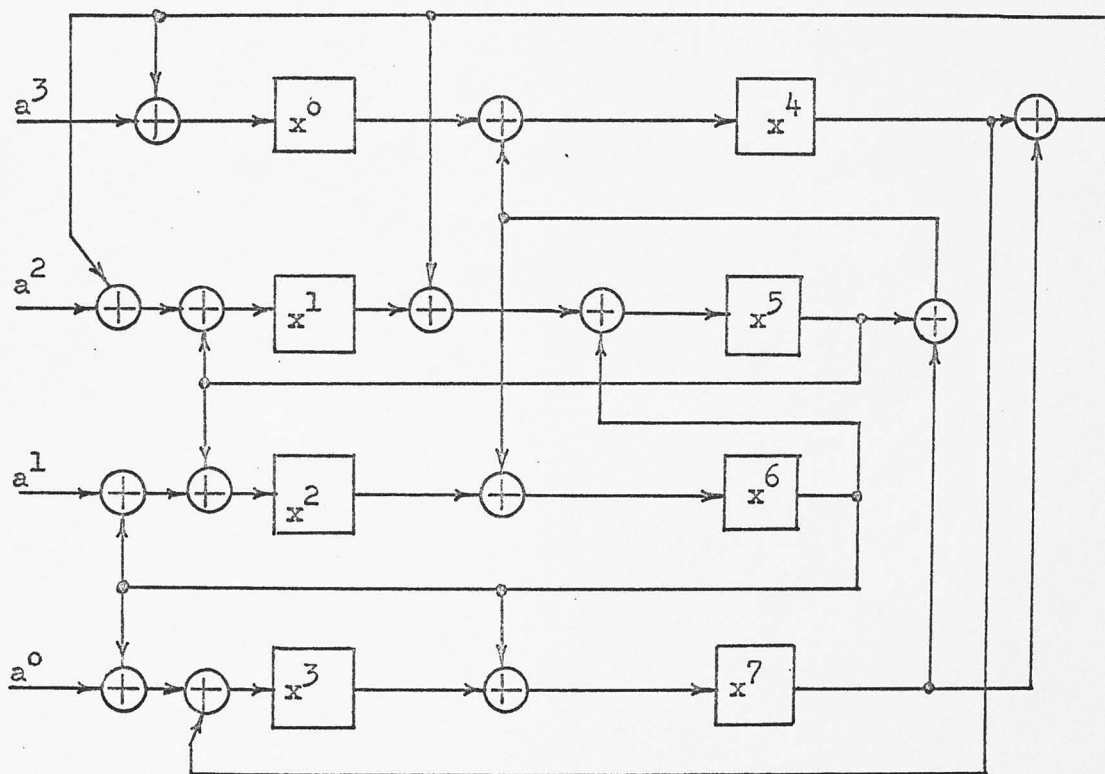


Figure 6.9 4-Channel Analogy GF Divider $[g_1(x)]_1$
 $g_1(x) = x^8 + x^5 + x^3 + x + 1$

The $[g_1(x)]_1$ is shown in Figure 6.9 and the parallel mode of the $[g_1(x)]_2$ is obtained from Figure 6.9 by having only one input line to the x^0 -position. This circuit is shown in Figure 6.10.

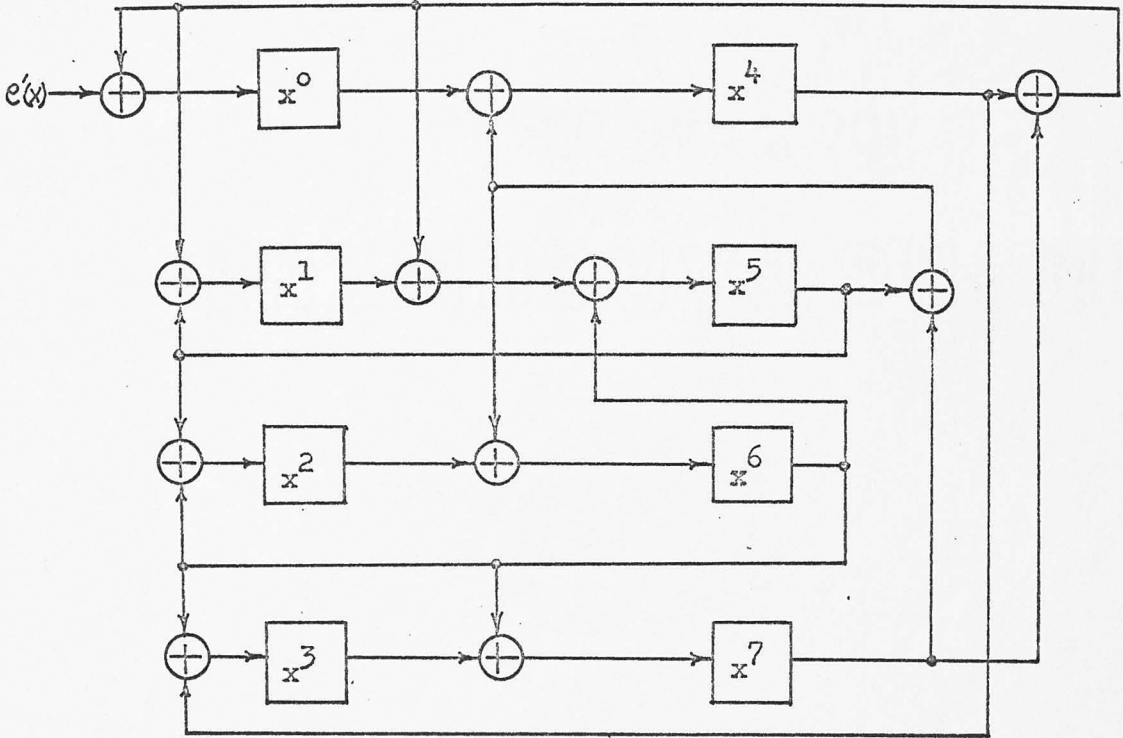


Figure 6.10 Parallel Mode of $[g_1(x)]_2$
 $g_1(x) = x^8 + x^5 + x^3 + x + 1$

Next, the serial mode of the $[g_1(x)]_2$ is shown in Figure 6.11.

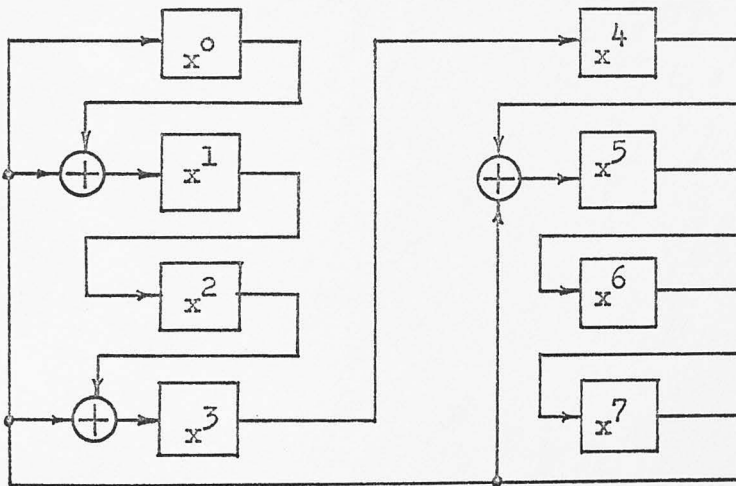


Figure 6.11 Serial Mode of $[g_1(x)]_2$
 $g_1(x) = x^8 + x^5 + x^3 + x + 1$

Table 6.1

	t	Input to $[g_1(x)]_1$	$[g_1(x)]_1$	Input	$[g_1(x)]_2$ Parallel mode
		$a^0 a^1 a^2 a^3$	$x^0 x^1 x^2 x^3 x^4 x^5 x^6 x^7$	$e'(x)$	$x^0 x^1 x^2 x^3 x^4 x^5 x^6 x^7$
Detection Cycle	0		0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0
	1	1 0 1 1	1 1 0 1 0 0 0 0	0	0 0 0 0 0 0 0 0
	2	1 0 1 0	0 1 0 1 1 1 0 1	1	1 0 0 0 0 0 0 0
	3	0 1 1 0	0 0 0 1 0 1 0 1	1	1 0 0 0 1 0 0 0
	4	1 0 0 0	1 0 1 1 0 1 0 1	1	0 1 0 1 1 1 0 0
	5	1 1 0 1	0 0 0 1 1 1 1 1	0	1 0 1 1 1 0 1 1
	6	0 1 1 1	1 0 1 0 0 1 0 0	1	1 0 1 0 0 1 0 0
	7	1 0 1 0	$r_1(x) =$ 0 0 1 1 0 0 0 0	0 $r_2(x) =$	0 1 1 0 0 0 0 0
Compari- son Cycle	t	Inputs to $[g_1(x)]_1$	$[g_1(x)]_1$	Input	$[g_1(x)]_2$ Serial mode
			$x^0 x^1 x^2 x^3 x^4 x^5 x^6 x^7$	$e'(x)$	$x^0 x^1 x^2 x^3 x^4 x^5 x^6 x^7$
	0		0 0 1 1 0 0 0 0	-	0 1 1 0 0 0 0 0
	1			match →	0 0 1 1 0 0 0 0
	2				
	3				
	4				
	⋮				

Suppose that errors are in the 4th channel, it is necessary to show that $j=1$ since $f=5$. From Figure 6.8, we have

$$\begin{aligned} e(x) &= x^{21} + x^{17} + x^{13} + x^5 = x^5 (x^{16} + x^{12} + x^8 + 1) \\ &= x^{\ell} E(x) \end{aligned}$$

Therefore,

$$E(x) = 1 + x^8 + x^{12} + x^{16}, \quad \ell=5$$

and also,

$$e'(x) = x^{\ell-j} E(x) = x^{5-j} (1 + x^8 + x^{12} + x^{16})$$

In the detection cycle, both $[g_1(x)]_1$ and $[g_1(x)]_2$ are 4-channel analogy GF dividers. The advancing of their states is shown in Table 6.1. Note that the states transition during the comparison cycle is also included. After all input bits are fed into the decoder, the remainder in $[g_1(x)]_1$ and $[g_1(x)]_2$ are $r_1(x)$ and $r_2(x)$ respectively. Here $r_1(x) \neq 0$ and $r_2(x) \neq 0$ because of the erroneous input data block. Next, the comparison cycle starts by fixing the content of the $[g_1(x)]_1$ and switching the $[g_1(x)]_2$ to its serial mode as shown in Figure 6.11. It can be seen from the comparison cycle of the Table 6.1 that after one time of serial feedback shift in $[g_1(x)]_2$ its content matches exactly that of the $[g_1(x)]_1$ and thus $j=1$ is determined.

The result can be checked mathematically. Since $j=1$, we have

$$e'(x) = x^4 (1 + x^8 + x^{12} + x^{16}) = x^4 + x^{12} + x^{16} + x^{20}$$

Then by long division, we have

$$\begin{aligned} x^{21} + x^{17} + x^{13} + x^5 &= (x^{13} + x^{10} + x^9 + x^8 + x^7 + x^2) \cdot (x^8 + x^5 + x^3 + x + 1) \\ e(x) \uparrow &\quad \quad \quad + (x^3 + x^2) r_1(x) \\ x^{20} + x^{16} + x^{12} + x^4 &= (x^{12} + x^9 + x^8 + x^7 + x^6 + x) \cdot (x^8 + x^5 + x^3 + x + 1) \\ e'(x) \uparrow &\quad \quad \quad + (x^2 + x) r_2(x) \end{aligned}$$

By comparing the above two equations, we have

$$r_1(x) \equiv r_2(x) \cdot x \quad \text{mod } g_1(x)$$

as it should be.

The actual error correction is then performed in the retransmission cycle. A bit in the 4th channel is converted whenever there is a 1 out of the decoder's mod 2 adder in this bit pulse time.

6.5 Reverse transmission and reciprocal polynomial

If the data in reverse transmission (process backward) is desired, then the $g_1(x)$ is best chosen to be a reciprocal polynomial. In this way, the code will preserve the same capability in error detection and error correction and hardware implementation. The following Theorem 6.2 was previously proven by Hsieh and the author. [42]

Theorem 6.2 Define the reciprocal polynomial $f^*(x)$ of any polynomial $f(x)$ to be $f^*(x) = x^m f(\frac{1}{x})$, where m is the degree of $f(x)$, then

1. The polynomial $f^*(x)$ is irreducible if and only if $f(x)$ is.
2. If $f(x)$ is irreducible, $f(x)$ and $f^*(x)$ belong to the same exponent. Therefore, $f^*(x)$ is primitive if and only if $f(x)$ is also.

Proof: See Hsieh and Hsiao. [42] Note that this theorem is a homework problem in Peterson. [3]

The following Theorem 6.3 states why a reciprocal polynomial is needed for reverse processing.

Theorem 6.3 In the "single channel error correction in a multi-channel transmission system," A 2-DVL code will have the same error detection and correction capability and the same amount of hardware implementation in reverse process provided that the chosen $g_1(x)$ is a reciprocal polynomial.

Proof:

$$F(x) = v(x) \cdot x^r + r(x) = q(x) \cdot g_1(x)$$

If the process is to be reversed, then we shall have $F^*(x)$ instead of $F(x)$, i.e.,

$$F^*(x) = \left(v(x) \cdot x^r + r(x) \right)^* = \left(q(x) \cdot g_1(x) \right)^*$$

By the definition of the reciprocal polynomial,

$$\begin{aligned} \left(q(x) \cdot g_1(x) \right)^* &= x^n q\left(\frac{1}{x}\right) \cdot g_1\left(\frac{1}{x}\right) = x^{n-r} q\left(\frac{1}{x}\right) \cdot x^r g_1\left(\frac{1}{x}\right) \\ &= q^*(x) \cdot g_1^*(x) \end{aligned} \quad (6.9)$$

Therefore, if $g_1(x)$ is a reciprocal polynomial,

$$g_1(x) = g_1^*(x)$$

then,

$$F^*(x) = q^*(x) \cdot g_1(x) \quad (6.10)$$

Equation (6.10) implies that the reverse process will preserve the same error detecting or correcting ability as that of the forward process in the system since the generator polynomial is the same. Furthermore, the hardware implementation of the system following the discussions given in previous sections is solely dependent on $g_1(x)$. Therefore, the amount of hardware used will be theoretically the same. Note also that $g_2(x) = 1 + x$ operates on columns and hence is independent of the process direction.

6.6 Comments on Brown and Sellers' scheme

D. T. Brown and F. F. Sellers of IBM invented a system and called it the cyclic redundancy check (CRC) which is used in the IBM 360/2400 tape series. ^[43] Their scheme is able to correct all errors in any one of the nine channels within a block. Their code has nine check bits to form a check character at the end of each block. Among the nine channels, one channel is the parity bit channel because every character (column) has

a parity bit. Error correction is then done by identifying the erroneous channel and retransmitting the whole block.

The generator polynomial of their code is

$$g(x) = 1 + x^3 + x^4 + x^6 + x^9$$

the circuit implementation of the above equation is shown in Figure 6.12.

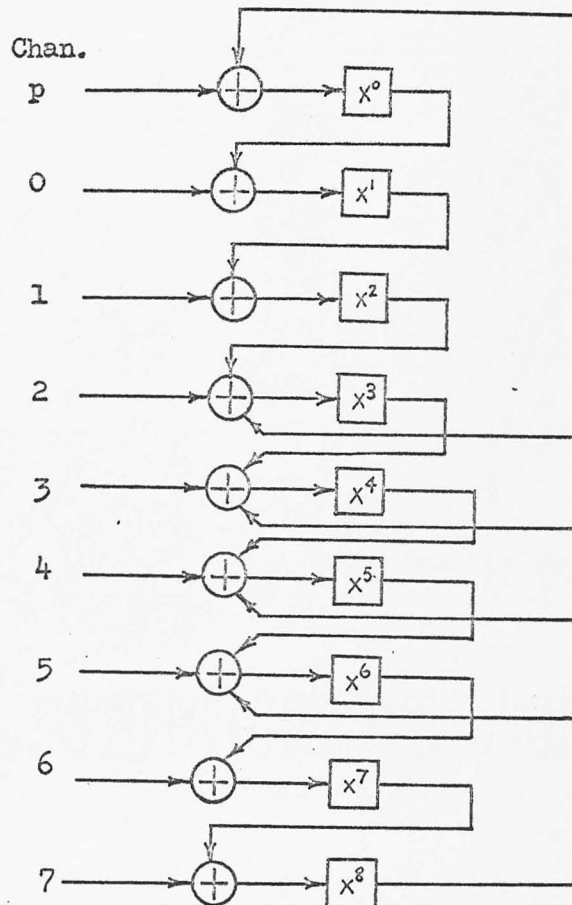


Figure 6.12

Though the circuit of Figure 6.12 has the parallel form, it is essentially a serial circuit because it is constructed according to the matrix T instead of T^F . More specifically, its characterizing matrix equation is of the following form.

$$S_{t+1} = S_t T + U_t \cdot G$$

where

$$T = \begin{bmatrix} 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \\ 10011110 \end{bmatrix} \quad G = \begin{bmatrix} 00000001 \\ 00000010 \\ 00000100 \\ 00001000 \\ 00010000 \\ 00100000 \\ 01000000 \\ 10000000 \end{bmatrix}$$

The circuit thus constructed has to make the number of channels equal to the degree of the generator polynomial. This restriction limits the error detection capability, hence the error correcting capability of the system. The generalized result given in this chapter based on a formal theory of parallel LFSR removes the above restriction. Furthermore, the code used in this chapter is the cyclic code, not as the modified cyclic code used by Brown and Sellers. This implies that the error detecting ability can be evaluated based on the existing result of the cyclic code. [44]

CHAPTER 7

PARALLEL GENERATION OF PN SEQUENCES

7.1 Generating PN sequences in parallel

Classic methods of generating PN sequences by a r -stage LFSR are done in the serial form, i.e., one bit at a time. Detailed study on the existing result may be referred to the books by Golomb, [13] Golomb, [14] Kautz, [12] and Peterson. [3] However, there is at least one report on generating PN sequences in parallel by using combinational network. [45]

In general, a PN sequence generated completely in parallel needs a shift-register of $2^r - 1$ stages. The number of shifts required in the worst case is $2^r - 2$. In this chapter, a new method is found for generating PN sequences of length $2^r - 1$ by a r -stage LFSR in a speed r times faster than the existing method and implementation hardware is increased only fractionally.

Definition 7.1: A PN sequence is defined to be a maximum-length linear recurring sequence modulo 2. This is, $\{a^k\}$ is a PN sequence if and only if it is a binary sequence which satisfies a linear recurrence

$$a^k = \sum_{i=1}^r g_i a^{k-i} \quad \text{mod } 2 \quad (7.1)$$

and has period $2^r - 1$. The number r is referred to as the degree of the PN sequence $\{a^k\}$. The polynomial

$$g(x) = 1 + \sum_{i=1}^r g_i x^i \quad (7.2)$$

is called the characteristic polynomial of the sequence $\{a^k\}$ of Equation (7.1). The above definition is adapted from Golomb. [13]

Usually, there are two ways of generating a PN sequence by using LFSR serially. That is, the chosen polynomial $g(x)$ of degree r must be primitive which can be implemented in two ways, as a serial GF divider or as a serial SRSR. However, only the later case is directly transformable

to its r -channel analogy circuit that will achieve a speed of generating the same PN sequence r times faster. The synthesis technique is simple because it is the autonomous case and only Equation (2.2.c) is needed. The result is now stated in Theorem 7.1.

Theorem 7.1: Let T be the companion matrix of a primitive polynomial $g(x)$ of degree r implemented as a SRSG, then the circuit implemented according to the connection matrix T^r will produce the same PN sequences of length 2^r-1 in a speed r times faster than the circuit implemented according to T , provided that r and 2^r-1 are relative prime.

Proof: The proof is divided into three parts.

1. Since the SRSG is implemented according to T^r and is autonomous, then

$$S'_{t+1} = T^r S'_t \quad (7.3)$$

Equation (7.3) is exactly a r -channel analogy SRSG under the conditions that $r=f$ and $U_t=0$. Therefore, it will produce the same output sequences as that of the serial SRSG.

2. In the parallel SRSG, the output sequence is produced r -bits a pulse time. This can be achieved by fetching out the content of each state which is r -bits. To show that this is true, the general circuit of a serial SRSG as shown in Figure 7.1 is needed.

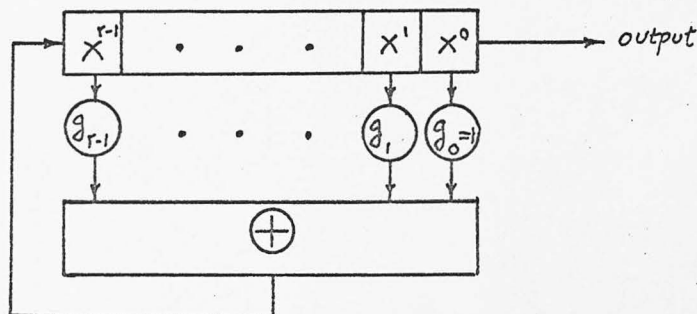


Figure 7.1 The Serial SRSG Characterized By $g(x)$

The outputs at time $i, i+1, \dots, i + (r-1)$ are as follows.

$$\left. \begin{aligned} b_i^0 &= x_i^0 \\ b_{i+1}^0 &= x_{i+1}^0 = x_i^1 \\ &\vdots \\ b_{i+r-1}^0 &= x_{i+r-1}^0 = x_{i+r-2}^1 = \dots = x_i^{r-1} \end{aligned} \right\} \quad (7.4)$$

Equation (7.4) implies that the r consecutive outputs of the SRSF are the contents of $x_i^0, x_i^1, \dots, x_i^{r-1}$. In a r -channel analogy SRSF, the state transition from S_{i-1} to S_i makes the content of the circuit changed as follows.

$$S_{i-1} \rightarrow S_i \Rightarrow (x_{i-1}^0, x_{i-1}^1, \dots, x_{i-1}^{r-1}) \rightarrow (x_i^0, x_i^1, \dots, x_i^{r-1})$$

i.e.,

$$S_{i-1} \rightarrow S_i \Rightarrow (x_{i-r+1}^0, x_{i-r+2}^0, \dots, x_{i-1}^0) \rightarrow (x_i^0, x_{i+1}^0, \dots, x_{i+r-1}^0)$$

Therefore, all r output bits are obtained once from each state content of the r -channel analogy SRSF circuit.

3. The reason for integers r and 2^r-1 being relative prime is to assure that the r -channel analogy SRSF has the same period of 2^r-1 . This result is stated in Chapter 4. If the period of the T^r circuit is less than 2^r-1 , then it will not be able to generate all possible PN sequences. The first positive integer r for which r and 2^r-1 are not relative prime is $r=6$. Other values of r can be easily found.

Note that in the above proof, no specific property of a PN sequence is used. Therefore, the theorem can be read more generally as follows. "A r -channel analogy SRSF will produce the same sequences as its original serial SRSF in a speed r times faster provided that r and the period of the serial SRSF are relative prime."

As stated before, a r -channel analogy GF divider cannot be used

directly to generate the same PN sequences r times faster than its original serial GF divider. The reason is that the second part of the proof of Theorem 7.1 does not hold any more.

7.2 A detail example

Example 7.1: Given $g(x) = 1 + x + x^4$, find a circuit that generates all PN sequences 4-times faster than the serial implementation of $g(x)$.

Solution: The serial SRSG is shown in Figure 7.2.

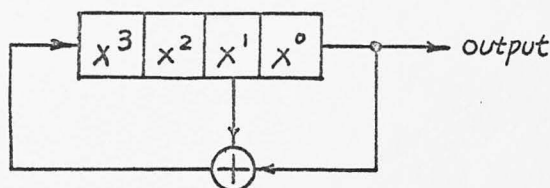
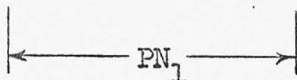


Figure 7.2 SRSG for $g(x) = 1 + x + x^4$

The succession of states with the initial state 0001 is listed in Table 7.1 and the output sequences is a PN sequence of the following form.

0001 0011 0101 1110 0010 0110 ...



By examining the above sequence 4-bit a time, it is noted that they are equivalent to the state transition $S_1 \rightarrow S_5 \rightarrow S_9 \rightarrow S_{13} \dots$ with the stages of the SRSG labelled as follows.

x^0	x^1	x^2	x^3	x^0	x^1	x^2	x^3
0	0	0	1	0	0	1	1...

The above analysis provides more insight to the problem. The 4-channel analogy SRSG is then constructed.

Table 7.1

States Transition of the Serial SRSG

t		state				output = x^0
		x^3	x^2	x^1	x^0	
1	s_1	1	0	0	0	0
2	s_2	0	1	0	0	0
3	s_3	0	0	1	0	0
4	s_4	1	0	0	1	1
5	s_5	1	1	0	0	0
6	s_6	0	1	1	0	0
7	s_7	1	0	1	1	1
8	s_8	0	1	0	1	1
9	s_9	1	0	1	0	0
10	s_{10}	1	1	0	1	1
11	s_{11}	1	1	1	0	0
12	s_{12}	1	1	1	1	1
13	s_{13}	0	1	1	1	1
14	s_{14}	0	0	1	1	1
15	s_{15}	0	0	0	1	1
16	s_{16}	1	0	0	0	0

$$T^4 = \begin{bmatrix} 1100 \\ 0110 \\ 0011 \\ 1101 \end{bmatrix}$$

The circuit is now shown in Figure 7.3.

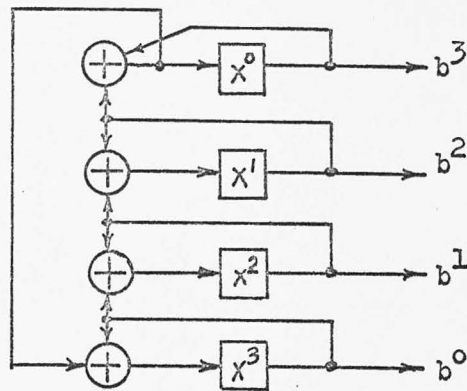


Figure 7.3 An Autonomous 4-Channel Analogy SRSG

Actually, as stated in Theorem 7.1, the circuit of Figure 7.3 will be able to generate all PN sequences in a speed 4-times faster than the circuit of Figure 7.2. However, for analysis purpose, a state transition table of the circuit of Figure 7.3 is given in Table 7.2.

Table 7.2 State Transition of the Circuit of Figure 7.3

t		x^3 x^2 x^1 x^0	Output = x^3 x^2 x^1 x^0
1	s_1	1 0 0 0	1 0 0 0
2	s_2	1 1 0 0	1 1 0 0
3	s_3	1 0 1 0	1 0 1 0
4	s_4	0 1 1 1	0 1 1 1
5	s_5	0 1 0 0	.
6	s_6	0 1 1 0	.
7	s_7	1 1 0 1	.
8	s_8	0 0 1 1	
9	s_9	0 0 1 0	
10	s_{10}	1 0 1 1	
11	s_{11}	1 1 1 0	
12	s_{12}	0 0 0 1	
13	s_{13}	1 0 0 1	
14	s_{14}	0 1 0 1	
15	s_{15}	1 1 1 1	
16	s_{16}	1 0 0 0	

In Table 7.2, the whole PN sequence is obtained within 4 shifting clock-time. Next, it is interesting to show that any PN sequence can be generated within 4 shifting clock-time. The result is compiled as follows.

Table 7.3

All possible PN sequences of length 15	States transition of the T^4 circuit
1 0001 0011 0101 111	$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4$
2 1000 1001 1010 111	$S_{12} \rightarrow S_{13} \rightarrow S_{14} \rightarrow S_{15}$
3 1100 0100 1101 011	$S_8 \rightarrow S_9 \rightarrow S_{10} \rightarrow S_{11}$
4 1110 0010 0110 101	$S_4 \rightarrow S_5 \rightarrow S_6 \rightarrow S_7$
5 1111 0001 0011 010	$S_{15} \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$
6 0111 1000 1001 101	$S_{11} \rightarrow S_{12} \rightarrow S_{13} \rightarrow S_{14}$
7 1011 1100 0100 110	$S_7 \rightarrow S_8 \rightarrow S_9 \rightarrow S_{10}$
8 0101 1110 0010 011	$S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6$
9 1010 1111 0001 001	$S_{14} \rightarrow S_{15} \rightarrow S_1 \rightarrow S_2$
10 1101 0111 1000 100	$S_{10} \rightarrow S_{11} \rightarrow S_{12} \rightarrow S_{13}$
11 0110 1011 1100 010	$S_6 \rightarrow S_7 \rightarrow S_8 \rightarrow S_9$
12 0011 0101 1110 001	$S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5$
13 1001 1010 1111 000	$S_{13} \rightarrow S_{14} \rightarrow S_{15} \rightarrow S_1$
14 0100 1101 0111 100	$S_9 \rightarrow S_{10} \rightarrow S_{11} \rightarrow S_{12}$
15 0010 0110 1011 110	$S_5 \rightarrow S_6 \rightarrow S_7 \rightarrow S_8$

7.3 Basic theorems on the structure of the parallel PN sequences

The parallel PN sequences generated by the method of Section 7.1 have some interesting algebraic properties same as the serial PN sequences.

Definition 7.2: A r -channel block sequence of length $n = cr + z$, (c, z are positive integers and $0 \leq z < r$) is defined as a binary two-dimensional sequence σ_i^j where the subscript designates the column number and the superscript designates the row number.

Since z is not necessary zero, the general expression of a r -channel block sequence Ψ is as follows:

$$\Psi = \sum_{j=1}^r \sum_{i=1}^c \sigma_i^j \cdot \sum_{j=1}^z \sigma_{c+1}^j$$

where the symbol "." is the concatenation. $X.Y$ means that X is followed by Y .

Example 7.2: The 4-channel block sequence of length 15 is expressed as

$$\sum_{j=1}^4 \sum_{i=1}^3 \sigma_i^j \cdot \sum_{j=1}^3 \sigma_4^j \implies \begin{matrix} \sigma_4^1 & \sigma_3^1 & \sigma_2^1 & \sigma_1^1 \\ \sigma_4^2 & \sigma_3^2 & \sigma_2^2 & \sigma_1^2 \\ \sigma_4^3 & \sigma_3^3 & \sigma_2^3 & \sigma_1^3 \\ \sigma_3^4 & \sigma_2^4 & \sigma_1^4 & \end{matrix}$$

Definition 7.3: A r -channel block PN sequence is a r -channel block sequence of length $n = 2^r - 1$, and the reading of $\sigma_1^1 \sigma_1^2 \dots \sigma_{c+1}^z$ is a serial PN sequence. Let $\Psi_1, \Psi_2, \dots, \Psi_{2^r-1}$ be r -channel block PN sequences, where

$$\begin{aligned} \Psi_1 &= \sum_{j=1}^r \sum_{i=1}^c \sigma_i^j \cdot \sum_{j=1}^z \sigma_{c+1}^j \\ \Psi_2 &= \sum_{j=1}^r \sum_{i=1}^c \sigma_{i+1}^j \cdot \sum_{j=1}^z \sigma_{c+2}^j \\ &\vdots \\ \Psi_{2^r-1} &= \sum_{j=1}^r \sum_{i=1}^c \sigma_{i+2^r-1}^j \cdot \sum_{j=1}^z \sigma_{c+2^r}^j \end{aligned}$$

with

$$\Psi_0 = \sum_{j=1}^r \sum_{i=1}^c 0_i^j \cdot \sum_{j=1}^z 0_{c+1}^j \quad (\text{All 0 block})$$

and then the following theorem is true.

Theorem 7.2: The r -channel block PN sequences $\psi_1, \psi_2, \dots, \psi_{2^r-1}$ plus ψ_0 form an Abelian group with respect to the operation of termwise addition modulo 2 provided that r and 2^r-1 are relative prime.

Here the word termwise addition modulo 2 means that if

$$\begin{array}{l}
 \text{then} \quad \psi_i = \begin{array}{c} \sigma_{c+1}^1 \quad \sigma_c^1 \quad \dots \quad \sigma_1^1 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \sigma_{c+1}^r \quad \sigma_c^r \quad \dots \quad \sigma_1^r \end{array} \quad \text{and} \quad \psi_j = \begin{array}{c} \rho_{c+1}^1 \quad \rho_c^1 \quad \dots \quad \rho_1^1 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \rho_{c+1}^r \quad \rho_c^r \quad \dots \quad \rho_1^r \end{array} \\
 \\
 \psi_i + \psi_j \implies \begin{array}{c} (\sigma_{c+1}^1 + \rho_{c+1}^1) (\sigma_c^1 + \rho_c^1) \dots (\sigma_1^1 + \rho_1^1) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ (\sigma_{c+1}^r + \rho_{c+1}^r) (\sigma_c^r + \rho_c^r) \quad \quad (\sigma_1^r + \rho_1^r) \end{array}
 \end{array}$$

Proof: The proof is divided into two parts, the first part is necessary for the second part of the proof. The second part is essentially the same as Golomb's proof for the serial case. [13, p45]

1. It is necessary to show that $\psi_1, \psi_2, \dots, \psi_{2^r-1}, \psi_0$ are unique, i.e., $\psi_i \neq \psi_j$ for all $i \neq j$ and $0 \leq i, j \leq 2^r-1$. This is true since any ψ_i is determined by specified linear recurrence relation R from its first bits, i.e., the first column of ψ_i . However, the first column of ψ_i is exactly the content of the state S_i of the r -channel analogy SRSG. It has been shown in Chapter 4, that the period of the r -channel analogy SRSG is equal to 2^r-1 provided that r and 2^r-1 are relative prime. Under this condition, all 2^r-1 states $S_1, S_2, \dots, S_{2^r-1}$ are nonzero and distinct. The addition of the state containing all zero makes all 2^r possible states. Therefore, $\psi_1, \psi_2, \dots, \psi_{2^r-1}$ and ψ_0 are unique.
2. This part of the proof follows directly from Golomb. [13] For any i we have $\psi_i + \psi_0 = \psi_i$ and $\psi_i + \psi_i = \psi_0$. Thus ψ_0 is the 0 element of the group, and every ψ_i is its own inverse. Let R be the linear relation satisfied by all ψ_i ($i=1, \dots, 2^r-1$), then it is also satisfied by ψ_0 . Clearly $\psi_i + \psi_j$ is also satisfied by R since R is linear. Thus $\psi_i + \psi_j$ is determined by R from its first r terms. Any pattern of this r term still belongs to one of the 2^r states. Therefore,

$$\psi_i + \psi_j = \psi_k \quad (7.6)$$

for some k which implies that the closure property is also satisfied. The associative and commutative law are clearly satisfied.

The result of Equation (7.6) implies that the r -channel block PN sequence also has the "shift-and-add" or "cycle-and-add" property.

Corollary 7.1: Theorem 7.2 can be modified for the case that the length $n \neq 2^r - 1$, and $n \geq r$.

Proof:

- (1) If $r < n < 2^r - 1$, then the corollary is obviously satisfied since the linear recurrence relation R holds for all bits in the block PN sequence of length $r < n < 2^r - 1$. However, if $n=r$, it is then simply the initial state.
- (2) If $n > 2^r - 1$, say $n = (2^r - 1) + d$ where d is a positive integer and less than $2^r - 1$ as shown in Figure 7.4,

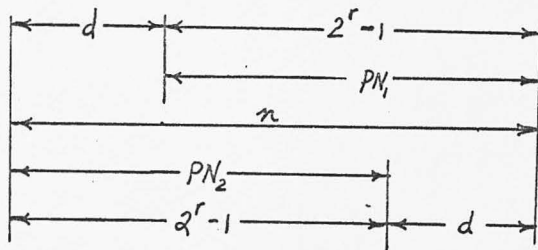


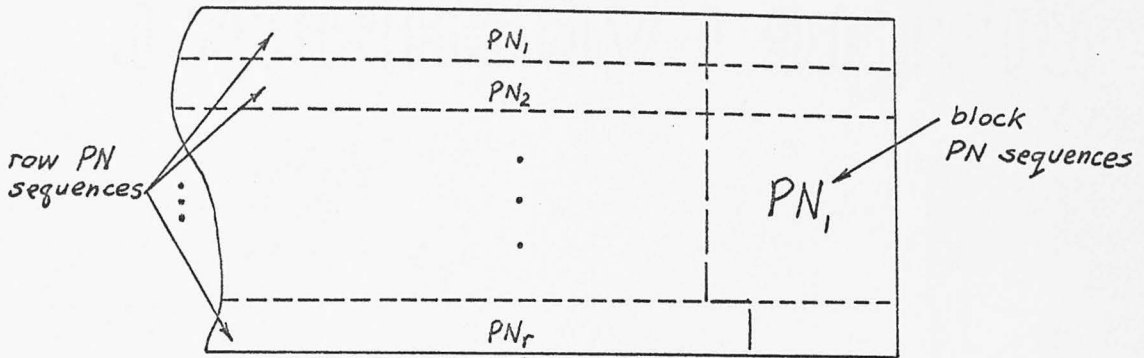
Figure 7.4

then the recurrence relation R satisfies PN_1 and PN_2 by definition. Moreover, R also satisfies d from (1). Therefore, R satisfies the concatenation of $PN \cdot d$ which has length n . The case of $d=1$ makes $n=2^r$ which implies that the block sequence is of rectangular form.

The next one of the theorems from Golomb is given here, which is needed to prove our next theorem.

Theorem 7.3: [Golomb 13] If $\{a^k\}$ is a PN sequence, then $\{a^{qk}\}$ equals $\{a^k\}$ except for a phase shift, when $q=1, 2, 4, 8, \dots, 2^{r-1}$.

Theorem 7.4: The sequences of length $2^r - 1$ generated from any x^i ($i = 0, 1, \dots, r-1$) position of a r -channel analogy SRSG in consecutive shifting times are also PN sequences provided that $r=2^i$ ($i \geq 1$, a positive integer) and is relative prime to $2^r - 1$. That is, we have the following sequence form, PN sequences in a row and block PN sequences.



Proof: In the case of serial SRSG, it is well known that the linear recurrence relation R satisfies both the output sequence and the sequence generated by any one stage of the SRSG relating the past contents of that stage with the future contents. [11] Therefore, a PN sequence $\{a^k\}$ is generated out of that stage of the serial SRSG. Next, the sequence generated by any one stage of the r -channel analogy SRSG is exactly $\{a^{rk}\}$. Then if $r=2^i$ for some i and r is relative prime to 2^r-1 , we have, from Theorem 7.3, that the sequence $\{a^{rk}\}$ generated is also a PN sequence.

Q.E.D.

The result of Example 7.1 is given here to illustrate the above stated theorems.

Example 7.3: Referring to Table 7.2, we have the following block PN sequences and row PN sequences shown in Figure 7.5.

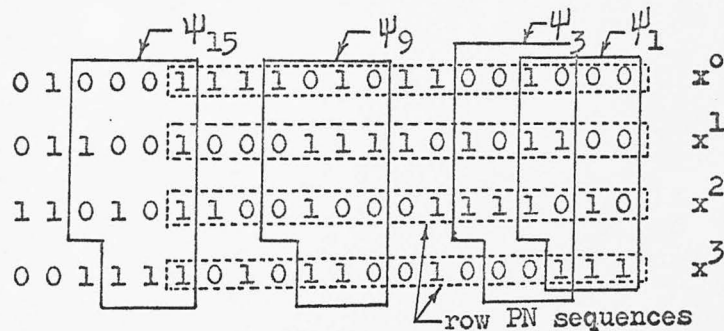


Figure 7.5

The block PN sequence ψ_1 , ψ_3 , ψ_9 and ψ_{15} are as indicated. The figure shows four row PN sequences. Other ψ 's can be labelled

accordingly. Note that the block PN sequences satisfy Theorems 7.2 and Corollary 7.1. The row PN sequences are a direct implication of Theorem 7.4 since here $r=4$ and it is relative prime to 15. The shift-and-add property of the block PN sequences can be easily checked, e.g.,

$$\psi_1 + \psi_3 = \psi_9 \quad (7.7)$$

Equation (7.7) still holds for $n=16$.

7.4 Correlation and orthogonality of the block sequence

In this section, the 2-dimensional autocorrelation function is defined. It is similar to conventional definition of 2-level autocorrelation in the one-dimensional case, i.e., the serial case. [13] [14]

In order to define the correlation function, the binary symbols 0 and 1 have to be changed to 1 and -1 respectively, i.e.

$$0 \rightarrow 1, \quad 1 \rightarrow -1$$

Definition 7.4: The cross-correlation ρ_{xy} of two equal length n r -channel block sequences X and Y is given by

$$\rho_{xy} = \frac{1}{n} \sum^n (X \times Y) \quad (7.8)$$

where the symbol $\sum^n (X \times Y)$ means that X is multiplied by Y in a termwise sense and then an arithmetic sum of all n termwise products is taken.

Example 7.4: Let X and Y be two 3×3 blocks where

$$X = \begin{matrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{matrix}, \quad Y = \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$$

then using the rules of $0 \rightarrow 1$ and $1 \rightarrow -1$, we have

$$\rho_{xy} = \frac{1}{9} (-6+3) = -\frac{1}{3}$$

Definition 7.5: The autocorrelation function $\rho_{\psi_1}(t)$ of a block sequence

ψ_i by shifting t columns is given by

$$\rho_{\psi_i}(t) = \frac{1}{n} \sum_{i=1}^n (\psi_i \times \psi_{i+t}) \quad (7.9)$$

Example 7.5: Using the ψ_1 of Figure 7.5, we have

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

and then $\rho_{\psi_1}(0) = 1$ and $\rho_{\psi_1}(1) = -\frac{1}{15}$ which are the same as the serial case. The autocorrelation function $\rho_{\psi_i}(t)$ may be calculated from the conventional form [13] [14]

$$\rho_{\psi_i}(t) = \frac{\# \text{ of disagreement} - \# \text{ of agreement}}{\# \text{ of disagreement} + \# \text{ of agreement}} \quad (7.10)$$

Definition 7.6: Two block sequences X and Y are said to be orthogonal if their termwise product sum is equal to zero, i.e., $\sum_{i=1}^n (X \times Y) = 0$.

Example 7.6: The following block sequences ψ_1 , ψ_2 and ψ_3 are orthogonal.

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

The orthogonality can be easily verified.

PROPOSED FUTURE RESEARCH

Future research on the subject of parallel feedback shift register can be divided into the following areas:

1). Parallel nonlinear feedback shift registers

- i). General synthesis techniques: References for the serial case are [13], [6], [4], [5] and [46].
- ii). Finite-state machine model and equivalent machines: References for the serial case are [13] and [17].
- iii). Periods and the generating function of a parallel nonlinear feedback shift register: References for the serial case are [13], [4] and [47].
- iv). Practical applications: Sequences generator, sequential machine design and other information processing units. References for the serial case are [13] and [14].

2). Parallel LFSR

a). Finite-state machine model

- i). Regular expression: The serial case has been worked by Brzozowski [48]. However, the problem for the parallel case is still unsolved. Its basic problem is due to the large number of input and output symbols.
- ii). Identification problem: Given different input block sequences, how can a parallel LFSR be identified? The nature of the problem in the serial case may be referred to p.177 of [31].

iii). Minimum machine: Method of finding a minimum machine is still unknown.

b). Applications

- i). Generation of parallel PN sequences of length other than $2^r - 1$. [13] [47]
- ii). Pattern recognition: The parallel nature of the parallel LFSR will be more suitable for dealing with the two-dimensional pattern recognition problem.
- iii). Data compression: Reference of using serial LFSR for data compression may be referred to Freiman and Chien. [49]
The nature of the general problem is in [50].
- iv). Decoding error correcting codes: Problems such as using parallel LFSR to implement BCH decoder are based on the Peterson-Chien algorithm [3] [39] and the algorithm suggested by Massey. The investigation of the possibility of implementing cyclic punctuate code [51] should be also studied.
- v). Radar range acquisition: The interwoven of block PN sequence and row PN sequence may suggest a new way of acquiring radar range very fast. References for the serial case are [52] [14] and [53].

APPENDIX 1.A.

Tables of elements of $GF(2^f)$, $f=2, 3, 4, 5, 6$

1. $f=2$, $g(x) = 1 + x + x^2$

ϕ 00 (ϕ is the zero element)

$\alpha^i, i=0$ 10 (α is a primitive root of $g(x)$)
 1 01
 2 11

2. $f=3$, $g(x) = 1 + x + x^3$

ϕ 000

$\alpha^i, i=0$ 100
 1 010
 2 001
 3 110
 4 011
 5 111
 6 101

3. $f=4$, $g(x) = 1 + x + x^4$

ϕ 0000

$\alpha^i, i=0$ 1000
 1 0100
 2 0010
 3 0001
 4 1100
 5 0110
 6 0011
 7 1101
 8 1010
 9 0101
 10 1110
 11 0111
 12 1111
 13 1011
 14 1001

4. $f=5, g(x) = 1 + x^2 + x^5$

ϕ 00000

$\alpha^i, i=$

0	10000
1	01000
2	00100
3	00010
4	00001
5	10100
6	01010
7	00101
8	10110
9	01011
10	10001
11	11100
12	01110
13	00111
14	10111
15	11111
16	11011
17	11001
18	11000
19	01100
20	00110
21	00011
22	10101
23	11110
24	01111
25	10011
26	11101
27	11010
28	01101
29	10010
30	01001

5. $f=6, g(x) = 1 + x + x^6$

ϕ 000000

$\alpha^i, i=$

0	100000
1	010000
2	001000
3	000100
4	000010
5	000001
6	110000
7	011000
8	001100
9	000110
10	000011
11	110001
12	101000
13	010100
14	001010
15	000101
16	110010
17	011001
18	111100
19	011110
20	001111
21	110111
22	101011
23	100101
24	100010
25	010001
26	111000
27	011100
28	001110
29	000111

30	110011
31	101001
32	100100
33	010010
34	001001
35	110100
36	011010
37	001101
38	110110
39	011011
40	111101
41	101110
42	010111
43	111011
44	101101
45	100110
46	010011
47	111001
48	101100
49	010110
50	001011
51	110101
52	101010
53	010101
54	111010
55	011101
56	111110
57	011111
58	111111
59	101111
60	100111
61	100011
62	100001

APPENDIX 1.B.

Miscellaneous tables for calculating T^f

1. $g(x) = x^8 + x^7 + x^6 + x^4 + 1$, Period = 15

α^i	i
0	10000000
1	01000000
2	00100000
3	00010000
4	00001000
5	00000100
6	00000010
7	00000001
8	10001011
9	11001110
10	01100111
11	10111000
12	01011100
13	00101110
14	00010111
	10000000

2. $g(x) = x^7 + x^6 + x^4 + 1$, Period = 15

α^i	i
0	10000000
1	01000000
2	00100000
3	00010000
4	00001000
5	00000100
6	00000010
7	10001011
8	11001111
9	11101110
10	01110111
11	10110000
12	01011100
13	00101110
14	00010111
	10000000

3. $g(x) = 1 + x + x^2 + x^4 + x^6$, Period = 21

a^i , i=	0	100000
	1	010000
	2	001000
	3	000100
	4	000010
	5	000001
	6	111010
	7	011101
	8	110100
	9	011010
	10	001101
	11	111100
	12	011110
	13	001111
	14	111101
	15	100100
	16	010010
	17	001001
	18	111110
	19	011111
	20	110101

REFERENCES

1. Hotz, G., "Mathematical Theory of Linear Sequential Networks in Switching Theory in Space Technology," Aiken and Main, Eds., Stanford University Press, 1963.
2. Cohn, M., "Properties of Linear Machines," J.ACM, Vol. 11, July 1964.
3. Peterson, W. W., "Error Correcting Codes," MIT Press, 1962.
4. Magleby, K. B., "The Synthesis of Nonlinear Feedback Shift Registers," Stanford Elec. Lab. TR 6207-1, Oct. 1963.
5. Yoeli, M., "Nonlinear Feedback Shift Registers," IBM TR 809, DSD Poughkeepsie, N.Y., Sept. 1961.
6. Golomb, S. Welch, L. R. and R. M. Goldstein, "Cycles from Nonlinear Shift Registers," JPL Report No. 20-389, August 1959.
7. Zierler, N., "Linear Recurring Sequences," SIAM Journal, Vol. 7, pp. 31-48, March 1959 (also in 12).
8. Zierler, N., "Several Binary Sequence Generators," MIT Lincoln Lab. TR No. 95, Sept. 1955 (also in 12).
9. Huffman, D. A., "The Synthesis of Linear Sequential Coding Networks in Information Theory," D. Cherry, editor, Academic Press, 1956 (also in 12).
10. Elspas, B., "Theory of Autonomous Linear Sequential Networks," IRE Trans. on Circuit Theory, March 1959 (also in 12).
11. Birdsall, T. G., and M. P. Ristenbatt, "Introduction to Linear Shift-Register Generated Sequences," EDG Technical Report 90, University of Michigan Research Institute, 1958.
12. Kautz, W. H., editor, "Linear Sequential Switching Circuits," Holden-Day, Inc., 1965.
13. Golomb, S., "Shift Register Sequences," Holden-Day, Inc., 1967.
14. Golomb, S., et al, "Digital Communications with Space Applications," Prentice Hall, 1964.
15. Brigham, R. C., "Some Properties of Binary Counters with Feedback," IRE Trans. on Elec. Computers, Dec. 1961.
16. Bryant, P. R., Heath, F. G. and R. D. Killick, "Counting with Feedback Shift Registers by Means of Jump Techniques," IRE Trans. on Elec. Computers, April 1962.

17. Guan Ji-wen, "The Theory of Singular Linear Autonomous Machines," ACTA Mathematica Sinica, Vol. 14, No. 5, 1962.
18. Huffman, D. A., "A Linear Circuit Viewpoint on Error Correcting Codes," IRE Trans. on Information Theory, Sept. 1956 (also in 12).
19. Kautz, W. H., "State Logic Relation in Autonomous Sequential Network," Proc. EJCC, Dec. 1958.
20. Perlman, M., "Implementation of Error Detection and Correction Using Binary Cyclic Codes," JPL TR 32-324, Oct. 1962.
21. Young, F. H., "Analysis of Shift Register Counters," J.ACM, Oct. 1958.
22. Hsiao, M. Y. and K. Y. Sih, "Serial-to-Parallel Transformation of Feedback Shift Register Circuits," IEEE Trans. on Elec. Computers, Dec. 1964.
23. Hsiao, M. Y. and K. Y. Sih, "Error Correction in Multi-Channel Circuits Employing Cyclic Codes," First IEEE Annual Communication Convention Record, Boulder, Colorado, June 1965.
24. Sih, K. Y. and M. Y. Hsiao, "Cyclic Codes in Multiple Channel Parallel System," IEEE Trans. on Elec. Computers, Dec. 1966.
25. Gill, A., "On the Serial-to-Parallel Transformation of Linear Sequential Circuits," IEEE Trans. on Elec. Computers, Feb. 1966.
26. Albert, A. A., "Fundamental Concepts of Higher Algebra," The University of Chicago Press, 1956.
27. Meggitt, J. E., "Error Correcting Codes and their Implementation," IRE Trans. on Information Theory, Oct. 1961.
28. Ash, R., "Information Theory," John Wiley and Son Co., 1965.
29. Melas, C. M., "A New Group of Codes for Correction of Dependent Errors in Data Transmission," IBM J. Res. & Dev., Jan. 1960.
30. Abramson, N. M., "Error Correcting Codes from Linear Sequential Networks," Proc., Fourth London Symposium on Information Theory, C. Cherry, Ed., Butterworths, Washington, D. C., 1961.
31. Gill, A., "Introduction to the Theory of Finite State Machines," McGraw-Hill, 1962.
32. Ginsburg, S., "An Introduction to the Mathematical Machine Theory," Addison-Wesley, 1962.
33. Roth, H. H., "Linear Binary Shift Register Circuits Utilizing a Minimum Number of Mod-2 Adders," IEEE Trans. on Information Theory, April 1965.

34. Perlis, S., "Theory of Matrices," Addison-Wesley Co., 1958.
35. Carter, W. C., "Linear Shift Register Sequence Generators and Systematic Codings of these Sequences," IBM SDD TR.1282, May 1965, Poughkeepsie, New York.
36. Preparata, F. P., "State-Logic Relations for Autonomous Sequential Networks," IEEE Trans. on Elec. Computers, Oct. 1964.
37. Fitzpatrick, G. B., "Synthesis of Binary Ring Counters of Given Period," J. ACM, July 1960.
38. Gantmacher, F. R., "The Theory of Matrices," Vol. 1, Chelsea Publishing Co., 1960.
39. Chien, R. T., "Cyclic Decoding Procedures for the BCH Codes," IEEE Trans. on Information Theory, Oct. 1964.
40. Marcus, M. P., "Switching Circuits for Engineers," Prentice-Hall, 1967.
41. Harrison, M. A., "Introduction to Switching and Automata Theory," McGraw-Hill, 1965.
42. Hsieh, P. and M. Y. Hsiao, "Solutions to Error Correcting Codes," (W. W. Peterson's book) IBM TR.00.999 and 1221, March 1963 and Dec. 1964.
43. Brown, D. T., "Error Correction in IBM 2400 Series Magnetic Tape," IBM TR.00.1244-1, March 1965, Poughkeepsie, N. Y.
44. Peterson, W. W., and D. T. Brown, "Cyclic Codes for Error Detection," Proc. IRE, Jan. 1961.
45. Anderson, T. O. and W. A. Lushbaugh, "Parallel Generation of the Check Bits of a PN Sequence," JPL Space Programs Summary No. 37-29, Vol. IV., 1963.
46. Mowle, F. J., "Enumeration and Classification of Stable Feedback Shift Registers," Dept. of EE, Univ. of Notre Dame, TR EE-661, Jan. 1966.
47. Perlman, M., "The Counting Task for Spacecraft Science Data Processing," JPL Technical Memo. 324-16, Feb. 1, 1967.
48. Brzozowski, J. A., "Regular Expressions for Linear Sequential Machine," IEEE Trans. on Elec. Computers, April 1965.
49. Freiman, C. V. and R. T. Chien, "Further Results in Polynomial Addressing," IBM J. of Res. and Dev., Oct. 1963.
50. Schwartz, J. W. and R. C. Barker, "Bit-Plane Encoding: A Technique for Source Encoding," IEEE Trans. on Aerospace and Electronic Systems, July 1966.

51. Solomon, G. and J. J. Stiffler, "Algebraically Punctured Cyclic Codes," Information and Control, Vol. 8, pp. 170-179, 1965.
52. Bartee, T. C. and P. E. Wood, "Coding for Tracking Radar Ranging," M.I.T. Lincoln Lab. TR 318, June 1963.
53. Bartee, T. C. and D. I. Schneider, "Computation with Finite Fields," Information and Control, Vol. 6, pp. 79-98, 1963.

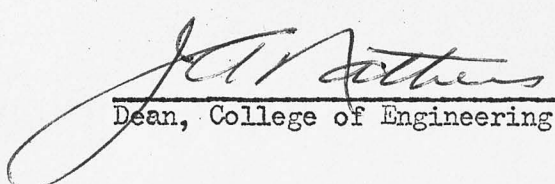
BIOGRAPHICAL SKETCH

Mu-Yue Hsiao was born July 17, 1933, in Changsha, Hunan, China. In June, 1952, he was graduated from Chung-Kung High School, Taipei, Taiwan. He received the degree of Bachelor of Electrical Engineering in June 1956, from National Taiwan University, Taipei, Taiwan. From 1956 to 1958, he served two years with the Chinese Air Force in Taiwan. In October, 1958, he came to the United States. He attended graduate school in the University of Illinois from 1959 to 1960 and received the degree Master of Science in Mathematics from that institution in June 1960. In September, 1960, he joined the International Business Machines Corporation, with whom he has been associated until August, 1965. He held several positions with that company and had assignments in various phases of digital computer development. He began work toward the degree of Doctor of Philosophy in September, 1965.

Mu-Yue Hsiao is married to the former Mona Yu-Chuan Shao and has two daughters. He is a member of the Institute of Electrical and Electronic Engineers, and Sigma Xi.

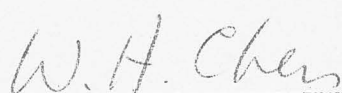
This dissertation was prepared under the direction of the chairman of the candidate's supervisory committee and has been approved by all members of that committee. It was submitted to the Dean of the College of Engineering and to the Graduate Council, and was approved as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1967


Dean, College of Engineering

Dean, Graduate School

Supervisory Committee


Chairman

