C H A P T E R VI

NONLINEAR FEEDBACK SHIFT REGISTER SEQUENCES

NONLINEAR FEEDBACK SHIFT REGISTER SEQUENCES

6.1 INTRODUCTION

A shift register of order n consists of n consecutive p-state (p 2) storage units regulated by a single clock. At each clock pulse, the state of each stage is shifted to the next stage in line. A shift register becomes a sequence generator by insertion of a feedback-loop, which computes a new term to the first stage based on the previous n terms. A shift register is called linear when the feedback logic includes only mod-p sums of the stored digits.

The device has an autonomous - as well as a forced mode. In autonomous mode, certain linear shift registers produce sequences with noiselike correlation properties, and this coupled with the simple and deterministic means of generation, makes them well-suited for use in system identification by correlation methods as shown in the previous chapters.

In this chapter, interest is confined to only the binary feedback shift registers (f.s.r.). In this case, out of 2^{2^n} feedback logics for n register elements, only 2^{n-1} give rise to linear sequences. The reminder of the available f.s.r. types are termed nonlinear, since the feedback logic must contain the nonlinear mod-2 multiplier element as well as mod-2 adder. The nonlinear régime makes a vast supply of new sequences available. In fact, the sequential behaviour of a general nonlinear f.s.r. still remains almost a complete mystery.

Although nonlinear f.s.r.s represent by far the largest section of available shift register types, not much of them has been documented. Early works in this area include those of Walker (1959), Golomb (1967), Mowle (1967), and Arvillias(1969).

Recently, Green et al (Jan.1970) described a technique of representing the feedback logic as a nonlinear polynomial using operations on a Karnaugh map, and used it to form polynomial products and analyze the sequential behaviour of product feedback shift register, (April,1970). Subsequently, Green et al (September 1970) has investigated the polynomial representation of the f.s.r. for its behaviour under the application of the dual and reverse operators.The concepts derived have been extended to include composite polynomial structures. However, many problems remained unsolved, some of which are the following :

Firstly, although quite promising, the use of Karnaugh map to obtain the characteristic polynomial of a nonlinear f.s.r. is somewhat involved,and this calls for a simple means of polynomial formation.

Secondly, Green et al (April,1970),in concluding his paper on nonlinear product feedback shift register,remarked that " A more difficult task in this field is the prediction of the cycle set of the composite f.s.r. from a knowledge of the cycle sets of the factor f.s.r.s. ...Apart from the general relationships mentioned, the exact nature of cycle-set formation has not yet been revealed, although extensive computations have indicated some definite form of the structure..."(page 686 of the paper).

Thirdly, as Green observes (April, 1970), the technique of Bryant et al (1967) for the description of the f.s.r.s that are linear except for a final inversion (Contralinear f.s.r.s) involves a sequential circuit consisting of a casecade of a 1-stage f.s.r. into the first f.s.r. with the final inversion removed. The contralinear f.s.r. is then viewed as a product feedback shift register. To formulate the cycle sets of such an f.s.r. a means of relating the sequences of the product f.s.r. to those of its factors is, therefore, necessary.

The objective of this chapter is to provide a satisfactory solution to each of the above three problems. The content of this chapter is presented as under :

Section 6.2 emphasizes the need for converting the feedback logic into a polynomial form. Section 6.3 gives basic relationships between the AND, OR, NOT logical connectives and mod-2 addition. Section 6.4 describes a simple and more direct method of finding the nonlinear characteristic polynomial from a knowledge of the f.s.r sequential behaviour. Section 6.5 presents a simple method of predicting the sequential behaviour of a composite f.s.r from a knowledge of its factors. Section 6.6 enumerates some useful theorems on the nonlinear f.s.r.s and their sequence domain consequences. Section 6.7 considers the analysis of a contralinear f.s.r viewing it as a product feedback shift register. Section 6.8 summarizes the salient features of the analysis of nonlinear feedback shift registers undertaken for study.

6.2 THE NEED FOR POLYNOMIAL FORM OF NONLINEAR FEEDBACK LOGIC

The following symbols are used throughout this chapter.

n = order of f.s.r.

x_i = ith stage in the register

1 = logical 1

$\oplus$ = modulo-2 addition

$*$ = modulo-2 multiplication (AND) (i.e. span increasing multiplication).

$\cdot$ = order increasing multiplication

x' = NOT $x = x \oplus 1$

$+$ = OR function

$F(x_1, x_2, .., x_n)$ = Boolean form of feedback function of n variables $x_1, x_2, \ldots, x_n$.

$F(x)$ = polynomial form of $F(x_1, x_2, .., x_n)$

$f_i(x)$ = factor polynomial

$RF(x)$ = reverse polynomial of $F(x)$

$DF(x)$ = dual polynomial of $F(x)$

The general configuration of a feedback shift register (f.s.r) of order n is shown in Fig. 6.1. Each binary storage element $X_1, X_2 .. X_n$ (delays the variable x_i on which it operates by one shi pulse period. At each shift pulse, the contents of X_i is transferre into X_{i+1}. The system is kept active by feeding some of the output: of the n stages into a logic device which depending on the logic function $F(x_1, x_2, .., x_n)$ provides input (0 or 1) to the first stage. On the application of a train of shift pulses, the f.s.r undergoes

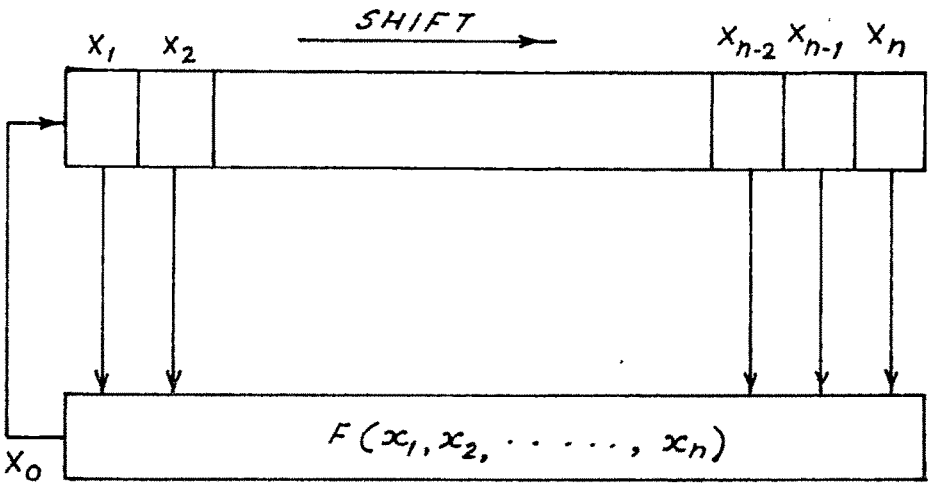


FIG. 6·1 GENERAL STRUCTURE FOR F.S.R. OF ORDER n

a cyclic succession of states, the number of which depends on the feedback logic function $F(x_1, x_2, .., x_n)$ and the initial state. Since the f.s.r. contains n stages, there is a total of 2^n distinct possible states. Hence, sequence generated can have a maximum period of 2^n digits.

The shifting action of the f.s.r may, therefore, be represented by the following equations :

$$\bar{x}_i = x_{i-1} \quad , \quad 1 \leq i \leq n \qquad \ldots \qquad \ldots \qquad (6.1)$$

and $$x_o = F(x_1, x_2, .. x_n) \qquad \ldots \qquad \ldots \qquad (6.2)$$

where $\bar{x}_i$ is the next value of x_i. Equation (6.2) describes the logic to determine the next input digit to the first stage of the f.s.r, which is either a '0' or a '1'. The feedback logic may, therefore, be expressed as a general Boolean equation in the logical connectives AND, OR and NOT.

For example, consider the repeating sequence :

1111010110010000, ... repeats

Characterizing this sequence by those states which are followed by a '1', the Boolean feedback logic function of the corresponding nonlinear f.s.r. is given by

$$\begin{aligned} F(x_1, x_2, x_3, x_4) &= (x_4 * x_3 * x_2 * x_1') + (x_4 * x_3' * x_2 * x_1') \\ &= + (x_4' * x_3 * x_2' * x_1) + (x_4 * x_3 * x_2' * x_1') \\ &\quad + (x_4' * x_3' * x_2' * x_1') + (x_4' * x_3' * x_2' * x_1) \\ &\quad + (x_4 * x_3' * x_2 * x_1) + (x_4' * x_3 * x_2 * x_1) \quad \ldots \quad (6.3) \end{aligned}$$

Apparently, the representation of nonlinear feedback logic as a general Boolean equation obscures the exact nature of nonlinearity. Also, such a representation is cumbersome and makes further manipulations difficult.

At this juncture, it is worthwhile recalling the fact that the effective study of linear feedback shift registers is largely dependent on a polynomial representation of the feedback logic. In the linear regime, this polynomial can be evolved directly from the feedback connections, and its composition determines the type of behaviour exhibited by f.s.r.

Viewing the present nonlinear situation in the light of the linear ohe, to set up a flexible theory of the nonlinear field,it is necessary to derive a polynomial form to operate in this regime. Once this polynomial form can be achieved, a number of manipulations become apparent.

To meet with the situation, Green et al (January, 1970) have broughtforth a means of converting the Boolean form of feedback logic into a polynomial form using operations on a Karnaugh map, and have shown that the polynomial form can furnish information on the sequential behaviour of the f.s.r which is not readily apparent from the original Boolean function. However, the procedure for obtaining the nonlinear polynomial making use of operations on a Karnaugh map appears to be somewhat involved and time consuming.

A more simple and direct method of forming the characteristic polynomial of a nonlinear f.s.r from a knowledge of its cycle structure is described in Section 6.4 of this chapter.

Some basic logical relationships that are of use throughout this chapter are written in the following section.

6.3 BASIC RELATIONSHIPS BETWEEN 'AND,' 'OR,' and 'NOT' FUNCTIONS and MOD-2 ADDITION

Since there is a direct correspondence between the suffix number of a register position (x_i) and the power to which the D operator of Huffman has to be raised (D^i) to represent the delay encountered at this stage, the former representation is used directly in a polynomial form of the feedback function.

The AND, OR, and NOT logical connectives are related to modulo-2 addition as :

$$x_i + x_j = (x_i' * x_j')' \quad \ldots \quad \ldots \quad (6.4)$$

$$x' = x \oplus 1 \quad \ldots \quad \ldots \quad (6.5)$$

When products involving nonlinear polynomials are considered, there are two types of multiplication :

(i) order increasing multiplication

(ii) span increasing multiplication

Span of a term means the number of variables joined by the AND operation. Of these two, the former is denoted by a dot and the latter by an asterisk.

Thus,

$$(x_i).(x_j) = (x_{i+j}) \qquad \text{order increasing} \quad \ldots (6.6)$$

$$(x_i)*(x_j) = (x_i * x_j) \qquad \text{span increasing} \quad \ldots (6.7)$$

Special cases -

$$(x_i).(x_i) = x_{2i} \quad \ldots \quad \ldots \quad (6.8)$$

$$(x_i)*(x_i) = x_i \quad \ldots \quad \ldots \quad (6.9)$$

Further relationships are as follows -

$$(x_i).(x_j \oplus x_k) = x_{i+j} \oplus x_{i+k} \quad \ldots \quad \ldots \quad (6.10)$$

$$(x_i)*(x_j \oplus x_k) = x_i * x_j \oplus x_i * x_k \quad \ldots \quad \ldots \quad (6.11)$$

$$(x_i \oplus x_j).(x_k \oplus x_m) = x_{i+k} \oplus x_{i+m} \oplus x_{j+k} \oplus x_{j+m} \quad (6.12)$$

$$(x_i \oplus x_j)*(x_k \oplus x_m) = x_i * x_k \oplus x_i * x_m \oplus x_j * x_k \oplus x_j * x_m \quad \ldots (6.13)$$

$$(x_i).(x_j * x_k) = x_{i+j} * x_{i+k} \quad \ldots \quad \ldots \quad (6.14)$$

$$(x_j * x_k).(x_i) = x_{i+j} * x_{i+k} \quad \ldots \quad \ldots \quad (6.15)$$

$$(x_i \oplus x_j).(x_k * x_m) = x_{i+k} * x_{i+m} \oplus x_{j+k} * x_{j+m} \quad \ldots \quad (6.16)$$

$$(x_k * x_m).(x_i \oplus x_j) = x_{i+k} * x_{i+m} \oplus x_{i+k} * x_{j+m} \oplus x_{j+k} * x_{i+m} \oplus x_{j+k} * x_{j+m} \quad \ldots \quad (6.17)$$

$$(1 \oplus x_i \oplus x_j \oplus \ldots \oplus x_n).f(x) = 1.f(x) \oplus x_i.f(x) \oplus x_j.f(x) \oplus \ldots \oplus x_n.f(x) \quad \ldots (6.18)$$

Further,

$$f_1(x).f_2(x) \neq f_2(x).f_1(x) \quad \ldots \quad \ldots \quad (6.19)$$

where either one or both the functions are nonlinear.

6.4 THE DIRECT FORMATION OF CHARACTERISTIC POLYNOMIAL OF A NONLINEAR f.s.r. FROM THE CYCLE STRUCTURE

A simple method is described here to form the characteristic polynomial of a nonlinear f.s.r from a knowledge of the shift register sequential behaviour (cycle structure).

Five properties of nonlinear f.s.r sequences that are of use in this section are given below. The first - and second of these properties N-1 and N-2 are rather evident. The third - and fifth properties N-3 and N-5 have not appeared in the literature. The fourth property N-4 applies to both linear as well as nonlinear sequences.

6.4.1 Five properties of nonlinear f.s.r sequences

Property N-1 : For an n-stage nonlinear f.s.r with 2^n different possible states, the maximum length of any generated sequence is 2^n digits. In this case, the register is called 'maximum length f.s.r.'

Property N-2 : If any n-consecutive digits (0s and 1s) are considered as a binary number between 0 and 2^n-1, all the binary numbers in this range appear once and once only in a maximum length nonlinear sequence.

Property N-2(a) : A sequence generated by an nth order nonlinear f.s.r consists of at the most n consecutive 0s or n consecutive 1s. This implies that the maximum run of 1s or 0s determines the order of the feedback logic function or equivalently number of stages in the f.s.r (i.e. n).

Property N-2(b): An n-stage feedback shift register generates a sequence consisting of n consecutive 1s and also n consecutive 0s provided the feedback logic is nonlinear. Alternatively, an n-stage linear f.s.r never generates a sequence with n consecutive 1s and also n consecutive 0s.

Property N-3 : There exists a linear relationship (in the modulo-2 sense) between any two or more consecutive states of a nonlinear f.s.r, excepting the case when the first of the two states is an 'all-zero'-state.

Example :

Consider the sequence generated by a 4-stage nonlinear f.s.r given by :

11110010000110, ... (repeats)

The states described by this f.s.r are as follows :

State No. of f.s.r	State of f.s.r (binary Number)
1 (initial)	1 1 1 1
2	0 1 1 1
3	0 0 1 1
4	1 0 0 1
5	0 1 0 0
6	0 0 1 0
7	0 0 0 1
8	0 0 0 0
9	1 0 0 0
10	1 1 0 0
11	0 1 1 0
12	1 0 1 1
13	1 1 0 1
14	1 1 1 0
15 = 1	1 1 1 1

For the sequence considered, possible linear (mod-2) relationships between two (or more) consecutive states of the corresponding f.s.r are written in tabular form below :

State No. of the f.s.r	A possible linear relation
13,14,1,2,3	$x_o = x_3 \oplus x_4$
3,4,5,6,7	$x_o = x_1 \oplus x_4$
7,8	$x_o = x_1 \oplus x_2$
8,9	No relation exists
9,10,11,12,13	$x_o = x_1 \oplus x_2$

Property N-3(a) : For any nonlinear f.s.r, there is a linear polynomial that describes maximum number of possible states governed by the nonlinear f.s.r. This property follows from the property N-3. For example, referring to the example cited above, the linear polynomial $x_o = x_1 \oplus x_2$ describes the maximum number of states of the non-linear f.s.r. Such a polynomial is accordingly termed as ' the best - fit linear polynomial' and denoted by the symbol $L(x)$. Thus in the present example $L(x) = x_1 \oplus x_2$.

Property N-4 : The mod-2 addition of a 1 to the output of the feedback logic device, derived from the contents of the kth - state, reverses only the output of the first stage in the $(k + 1)$th state.

This property follows directly from the shifting action of an f.s.r and the property of mod-2 addition operation.

Illustration of Property N-4 :

State No. of f.s.r	Next value of x_1	Stages of f.s.r		
	x_o	x_1	x_2	x_n
1. (initial) state	a_o	a_{-1}	a_{-2}	a_{-n}
2. state	a_1	a_o	a_{-1}	a_{-n+1}
.....	.....		.....	
.....	.....		.....	
k-th state (1 is added in the mod-2 sense to the f.s.r input (x_o) formed from k-th state)	$1 \oplus a_{k-1}$	a_{k-2}	a_{k-3}	a_{k-n-1}
(k+1)th state	a_k (computed from k+1th state)	$(1 \oplus a_{k-1})$	a_{k-2}	a_{k-n}

Property N-5 : A state of an n-stage f.s.r can be expressed as a logical product function of n variables, x_1 to x_n each occuring only once either in true or complemented form. When the variables of a state are all 1s, they can serve as inputs to an AND gate to compute a logical 1. When any of the variables of a state is a '0', by considering its complementary form, a new product function for the state can be formed which satisfies the AND condition. Such a product function may be called 'a switching function, s(x)' and can be used to compute a logical 1 corresponding to a state of the f.s.r. For instance, consider the state of a 4th order f.s.r : $x_1 = 0$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 0$. The switching function for this state is then, $s(x) = (x_1' * x_2' * x_3 * x_4')$

For example, consider the following two cases which describe a state of 4th order f.s.r. :

Case 1 : $x_1*x_2*x_3*x_4'$ which computes a logical 0 to the input of the register

Case 2 : $x_1*x_2*x_3*x_4$ which computes a logical 1 to the input of the register

The switching function in Case 1 is, therefore -

$s_1(x) = x_1'*x_2'*x_3'*x_4$

In case 2, the given state satisfies the AND condition and therefore this itself is the switching function i.e.

$s_2(x) = x_1*x_2*x_3*x_4$

6.4.2 The procedure for formulating the nonlinear polynomial

The steps involved in the formation of the nonlinear characteristic polynomial from the knowledge of the shift register cycle structure are the following :

Step 1 : Making use of the properties N-1, N-2 and N-2(a) and N-2(b), determine the value of n, the number of stages in the nonlinear f.s.r that generates the given sequence.

Step 2 : Write all the consecutive states of the f.s.r from the given sequence.

Step 3 : Examine the f.s.r states set out in Step 2, and, obtain a linear polynomial that describes any two or more consecutive states. (Property 3N-3). In case, the so obtained polynomial is the 'best-fit linear polynomial, L(x)' , the computation is simplified. (The term L(x) is explained in property N-3(a).)

Step 4 : Write all the states of the linear shift register characterized by the linear polynomial L(x).

Step 5 : Compare the states of the linear shift register with those of the nonlinear f.s.r, and obtain those states governed by the linear polynomial where a change (i.e. reversal since the system is binary) is required to make them (states) identical with the corresponding states of the nonlinear f.s.r.

Step 6 : Add a '1' in the mod-2 sense to the output of the feedback logic device derived from the kth state of the linear shift register when a change is required at the (k+1)-th state. Repeat this at all the positions where alterations are necessary (Property N-4).

The required '1', that is to be modulo-2 added to the output of the feedback logic device derived from k-th stage, can be easily obtained by decoding the k-th state using AND logic.

Step 7 : From properties N-4 and N-5, the modulo-2 addition of a '1' to the output digit of the feedback logic device formed from the k-th state of the linear shift register, means the modulo-2 addition of the switching function $s_k(x)$ to the linear polynomial L(x).

Each time a 1 is modulo-2 added, the corresponding switching function $s_i(x)$ must be modulo-2 added to the chosen linear polynomial L(x) to obtain the feedback logic function for the formed states of the f.s.r.

In mathematical terms, the above situation may be expressed as :

$$F(x_1, x_2, .., x_n) = L(x) \oplus \sum_{i=1}^{r} s_i(x) \quad , \quad 1 \leq r \leq 2^n - 1$$

where $F(x_1, x_2, .., x_n)$ is the feedback logic of the nonlinear f.s.r. corresponding to the given sequence, but it is present in a form involving mod-2 addition, mod-2 multiplication (AND), and inversion)(NOT).

Step 8 : By employing the transformation

$$x_i' = x_i \oplus 1 \quad ,$$

the switching function $s_i(x)$ takes on the form involving only mod-2 addition, and mod-2 multiplication operations. Thus,

$$F(x) = L(x) \oplus \left[\sum_{i=1}^{r} s_i(x) \right]_{x_i' = x_i \oplus 1} \quad \ldots \quad \ldots \quad (6.20)$$

is the required characteristic polynomial of the nonlinear feedback shift register.

Step 9 : Simplifying the eqn. (6.20) and arranging the terms in the order-increasing manner leads to the final desired form of the nonlinear characteristic polynomial F(x).

To clear the course of the analysis, two examples are illustrated below, one for maximal - and the other for nonmaximal nonlinear cycle structure.

Example 6.1 : (A case of maximal nonlinear sequence)

It is desired to form the feedback logic of the shift register generating the following sequence into a polynomial form :

1111010110010000, ... repeats

The solution to this problem is presented below in accordance with the steps stated earlier in this section.

Step 1 : The repetetion period of the given sequence is 16 digits. There are 4 consecutive 1s and also 4 consecutive 0s in it. Further, the sequence describes all the possible 4-digit distinct binary numbers from 0 to 15 exactly once in one period.

Hence, it can be concluded that the sequence is nonlinear and can be generated by a 4th order maximum length f.s.r. Thus,

$$n = 4$$

Step 2 : The states of the nonlinear f.s.r (i.e. the 4-digit consecutive binary numbers in the given sequence) are shown in Table 6.1, on the page 495.

Step 3 : A close inspection of the states of the nonlinear f.s.r. in Table 6.1 reveals that the corresponding best-fit linear polynomial, L(x), is given by :

$L(x) = x_1 \oplus x_4$ ... (i.e. modulo-2 sum of the 1st - and 4th stages)

Step 4 : The states of the linear shift register characterized by the linear polynomial L(x) are also stated in Table 6.1 to facilitate easy comparisons of linear and nonlinear situations.

Table 6.1 : Illustration of the Steps 2, 3, 4, 5 and 6 in forming the nonlinear polynomial F(x)

State Numbers	Nonlinear f.s.r					Best fit linear f.s.r					Best fit linear f.s.r after effecting mod-2 addition(s)				
	Next value of x_1	Stage Numbers				Next value of x_1	Stage Numbers				Next value of x_1	Stage Numbers			
	x_o	x_1	x_2	x_3	x_4	x_o	x_1	x_2	x_3	x_4	x_o	x_1	x_2	x_3	x_4
1	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1
2	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1
3	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1
4	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1
5	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0
6	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1
7	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
8	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1
9	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1
10	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
11	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
12	0	0	0	0	1	1	0	0	0	1	$1 \oplus 1 = 0$	0	0	0	1
13	1	0	0	0	0	1	1	0	0	0	$1 \oplus 0 = 1$	0	0	0	0
14	1	1	0	0	0	1	1	1	0	0	1	1	0	0	0
15	1	1	1	0	0	1	1	1	1	0	1	1	1	0	0
16	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0
17	0	1	1	1	1		... Repeats...				0	1	1	1	1
		...Repeats...										...Repeats...			

Steps 5 and 6 : Comparing the states of the best-fit linear f.s.r with those of the nonlinear f.s.r, it is clear that, at first, a change is required at the first stage of the linear f.s.r. at state No. 13.

A '1' is therefore added in the modulo-2 sense to the input digit of the linear f.s.r (x_o) derived from its 12th state. By this mod-2 addition, the state 13th in the linear f.s.r is made identical with the corresponding state in the nonlinear f.s.r.

But, when the above change is brought in, the linear f.s.r assumes the 'all-zero' state and so becomes insensitive to the linear modulo-2 logic.

Therefore, to realize the 14th state of the nonlinear f.s.r at the corresponding state of the linear f.s.r, it is once again necessary to add a '1' in the mod-2 sense) to the input digit of the linear f.s.r (x_o) derived from its 'all-zero' 13th state. With this addition, the linear f.s.r falls in line with the nonlinear f.s.r and hence describes the same states as in the nonlinear situations.

These steps are clearly shown in Table 6.1.

Hence, it is possible to realize a given nonlinear sequence starting from a best-fit linear sequence and affecting into this chosen linear sequence the necessary additions of a '1' in the modulo-2 sense at the required positions.

Step 7 : From the results obtained in the previous steps the nonlinear feedback logic function $F(x_1, x_2, x_3, x_4)$ involving mod-2 addition, mod-2 multiplication (AND) , and NOT operation is given by :

$$F(x_1, x_2, x_3, x_4) = L(x) \oplus \sum_{i=1}^{2} s_i(x)$$

X Here i = 1, and 2 as there are two mod-2 additions to the linear sequence as shown in Table 6.1

$$= x_1 \oplus x_4 \oplus s_1(x) \oplus s_2(x)$$

The switching function $s_1(x)$ corresponds to the state 12 of the linear f.s.r of Table 6.1 and is given by :

$$s_1(x) = (x_1' * x_2' * x_3' * x_4)$$

Similarly, the switching function $s_2(x)$ corresponds to the state 13 of the linear f.s.r of Table 6.1 and is given by :

$$s_2(x) = (x_1' * x_2' * x_3' * x_4')$$

Hence,

$$F(x_1, x_2, x_3, x_4) = (x_1 \oplus x_4 \oplus x_1' * x_2' * x_3' * x_4 \oplus x_1' * x_2' * x_3' * x_4')$$

Step 8 : By employing the transform :

$$x_i' = x_i \oplus 1$$

in $s_1(x)$ and $s_2(x)$, the NOT operation in them can be eliminated in which case the corresponding switching functions are denoted by $S_1(x)$ and $S_2(x)$ respectively and are given below :

$$S_1(x) = x_1' * x_2' * x_3' * x_4 \qquad x_i' = x_i \oplus 1$$

$$= (x_1 \oplus 1) * (x_2 \oplus 1) * (x_3 \oplus 1) * x_4$$

Likewise,

$$S_2(x) = [\, x_1'*x_2'*x_3'*x_4' \,]_{x_i' = x_i \oplus 1}$$

$$= (x_1 \oplus 1)*(x_2 \oplus 1)*(x_3 \oplus 1)*(x_4 \oplus 1)$$

Adding $S_1(x)$ and $S_2(x)$ in the mod-2 sense, and simplifying, thef following equation is obtained :

$$S_1(x) \oplus S_2(x) = (1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1*x_2 \oplus x_1*x_3 \oplus x_2*x_3 \oplus x_1*x_2*x_3)$$

Hence, the nonlinear characteristic polynomial involving only mod-2 addition and mod-2 multiplication is (eqn. 6.20)

$$F(x) = L(x) \oplus S_1(x) \oplus S_2(x)$$

$$=(x_1 \oplus x_4 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1*x_2 \oplus x_1*x_3 \oplus x_2*x_3 \oplus x_1*x_2*x_3)$$

Step 9 : Since $x_i \oplus x_i = 0$, (mod-2 addition property) the final expression for nonlinear feedback logic in polynomial form is :

$$F(x) = (1 \oplus x_2 \oplus x_1*x_2 \oplus x_3 \oplus x_1*x_3 \oplus x_2*x_3 \oplus x_1*x_2*x_3 \oplus x_4)$$

Example 6.2 : (A case of nonmaximal nonlinear sequence)

It is desired to form the feedback logic of the shift register generating the following sequence into a polynomial involving only mod-2 addition and mod-2 multiplication :

110000010101,... repeats

The solution to this problem is presented below in accordance with the steps set out earlier in this section, leading to the final desired form of the polynomial as stated in eqn. (6.20).

Step 1 : The repetetion period of the given sequence is 12 digits. There are five consecutive zeros in it. In view of properties N-1, N-2, N-2(a), and N-2(b) of nonlinear sequences stated earlier, it can be said that the above sequence is a nonlinear nonmaximal type, and can be generated by a sixth order feedback shift register. Thus,

$$n = 6$$

Step 2 : The states of the nonlinear f.s.r (6-digit consecutive binary numbers in the given sequence) are shown in Table 6.2.

Step 3 : A close inspection of the states of nonlinear f.s.r of Table 6.2 (given on the next page) reveals that the best-fit linear polynomial, L(x), (i.e. the polynomial that describes maximum number of states of the nonlinear f.s.r) is given by :

$$L(x) = x_2 \oplus x_5 \oplus x_6 \quad .. \text{ (mod-2 sum of 2nd, 5th and 6th stages)}$$

Step 4 : The states of the linear f.s.r governed by the polynomial L(x) are also written in Table 6.2 to facilitate ease in the comparison of the linear and nonlinear situations.

Table 6.2 : Illustration of the steps 2,3,4,5 and 6 in forming the nonlinear polynomial F(x)

State No.	Nonlinear f.s.r stages							Best-fit linear f.s.r stages							After effecting mod-2 addition linear f.s.r stages						
	$\overline{x_o}$	x_1	x_2	x_3	x_4	x_5	x_6	$\overline{x_o}$	x_1	x_2	x_3	x_4	x_5	x_6	$\overline{x_o}$	x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1
2	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1
3	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
4	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0
5	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0
6	1	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0
7	1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0	1	0
8	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1⊕0=1	1	1	0	1	0	1
9	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	1	0	1	0
10	0	0	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	1	1	0	1
11	0	0	0	1	1	1	0	1	1	0	0	1	1	0	1⊕1=0	0	0	1	1	1	0
12	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1

As before, in this Table too, x_o represents the next value of x_1

Steps 5 and 6 : On comparison of states of the linear f.s.r with those of the nonlinear (Table 6.2) it is seen that, at first, a change is required at the first stage of the linear f.s.r at state No.9.

A '1' is therefore added in the mod-2 sense to the input digit of the linear f.s.r (x_o) computed from its 8th state. By this linear addition, the state 9 of the linear f.s.r is made identical with the state 9 of the nonlinear f.s.r.

Also, the states 10, and 11 of the linear f.s.r as per the linear logic given by L(x) are found to agree with the corresponding states of the nonlinear f.s.r.

However, the state 12 of the linear f.s.r differs from that of the nonlinear in the first stage. To make this 12th state of linear f.s.r to fall in line with the 12th state of the nonlinear f.s.r, a '1' is modulo-2 added to the input digit of the linear f.s.r (x_o) computed from its 11th state.

In this manner, the states of the nonlinear f.s.r are derived. This situation speaks for the fact that it is always possible to arrive at a nonlinear sequence by starting from the corresponding best-fit linear sequence and modulo-2 adding a '1' at the desired locations.

These steps are clearly shown in Table 6.2.

Step 7 : From the results of the previous sixth steps, the nonlinear feedback function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ involving mod-2 addition and mod-2 multiplication (AND), and NOT operation is

given by :

$$F(x_1, x_2, \ldots, x_6) = L(x) \oplus \sum_{i=1}^{2} s_i(x)$$

Here i takes only two values because only 2 mod-2 additions of a '1' are made to the linear sequence as shown in Table 6.2

The switching function $s_1(x)$ corresponds to the 8th state of the linear f.s.r of Table 6.2 and is therefore given as :

$$s_1(x) = (x_1 * x_2 * x_3' * x_4 * x_5' * x_6)$$

Similarly, the switching function $s_2(x)$ corresponds to the 11th state of the linear f.s.r of Table 6.2 and is thus given by :

$$s_2(x) = (x_1' * x_2' * x_3 * x_4 * x_5 * x_6')$$

Hence,

$$F(x_1, x_2, x_3, x_4, x_5, x_6) = (x_2 \oplus x_5 \oplus x_6 \oplus x_1 * x_2 * x_3' * x_4 * x_5' * x_6 \oplus x_1' * x_2' * x_3 * x_4 * x_5 * x_6')$$

Step 8 : By employing the transform :

$$x_i' = x_i \oplus 1$$

The NOT operation in the switching functions $s_1(x)$ and $s_2(x)$ can be eliminated and the so transformed versions of $s_1(x)$ and $s_2(x)$ may be denoted by $S_1(x)$ and $S_2(x)$. Thus,

$$S_1(x) = [x_1 * x_2 * x_3' * x_4 * x_5' * x_6] \qquad x_i' = x_i \oplus 1$$

$$= x_1 * x_2 * (x_3 \oplus 1) * x_4 * (x_5 \oplus 1) * x_6$$

Similarly,

$$S_2(x) = (x_1 \oplus 1) * (x_2 \oplus 1) * x_3 * x_4 * x_5 * (x_6 \oplus 1)$$

Sum of $S_1(x)$ and $S_2(x)$ in the mod-2 sense may be expressed as :

$$\begin{aligned} S_1(x) \oplus S_2(x) = \; & (x_1*x_2*x_3*x_4*x_5) \oplus (x_1*x_2*x_3*x_4*x_6) \\ & \oplus (x_1*x_2*x_4*x_5*x_6) \oplus (x_1*x_3*x_4*x_5*x_6) \\ & \oplus (x_1*x_2*x_4*x_6) \oplus (x_1*x_3*x_4*x_5) \\ & \oplus (x_2*x_3*x_4*x_5) \oplus (x_3*x_4*x_5*x_6) \\ & \oplus (x_2*x_3*x_4*x_5*x_6) \oplus (x_3*x_4*x_5) \end{aligned}$$

At the first sight, this seems to be quite an involved equation. But, checking these AND functions with the n-digit numbers contained in the given nonlinear sequence (i.e. the states of the nonlinear f.s.r of Table 6.2), it is quite easy to see that except the two terms : $(x_1*x_2*x_4*x_6)$ and $(x_3*x_4*x_5)$ all others do not correspond to any possible state of the f.s.r and are therefore not of any present concern. Thus,

$$S_1(x) \oplus S_2(x) = (x_1*x_2*x_4*x_6) \oplus (x_3*x_4*x_5)$$

And, the required nonlinear feedback function in polynomial form may now be written as :

$$\begin{aligned} F(x) &= L(x) \oplus S_1(x) \oplus S_2(x) \\ &= (x_2 \oplus x_5 \oplus x_6 \oplus x_1*x_2*x_4*x_6 \oplus x_3*x_4*x_5) \end{aligned}$$

Step 9 : Rearranging the terms in the above equation leads to the final form of polynomial F(x) given by :

$$F(x) = (x_1*x_2*x_4*x_6 \oplus x_2 \oplus x_3*x_4*x_5 \oplus x_5 \oplus x_6)$$

Verification : Starting from any 6-digit number in the given sequence, the above function is seen to describe all the rest.

6.4.3 Discussion and further use of proposed method of polynomial formation

The method proposed here for obtaining a polynomial form of nonlinear feedback logic from a knowledge of the f.s.r cycle structure is quite simple and straight approach. It does not involve any complex mathematical transformations. Further, the method provides quick results, and is seen to give valid results irrespective of whether the given sequence is maximal ($= 2^n$ digits) or nonmaximal ($< 2^n$ digits).

The computations in the method are simplified by choosing the best-fit linear polynomial L(x), which can describe maximum number of possible nonlinear f.s.r states. However, the procedure yields the same final results whether the chosen polynomial L(x) is a best-fit or otherwise.

Also, it may be noted that, following the steps of the proposed method in its reverse order facilitates easy evaluation of a sequence associated with a given characteristic polynomial of a nonlinear f.s.r.

As the method starts from a linear sequence to derive the desired nonlinear sequence, the exact nature of the nonlinearity is explicitely known, thereby providing a better insight to the nonlinear problem.

6.5 PREDICTION OF THE CYCLE SETS OF A COMPOSITE NONLINEAR f.s.r. FROM A KNOWLEDGE OF THE FACTOR f.s.r.s.

In the linear regime, the composition of the characteristic polynomial determines the sequential behaviour of the f.s.r. Given a linear polynomial, irreducible or composite, it is possible to evaluate the associated cycle set. The development of this aspect is included previously in chapters 1 and 3 covering the binary- and p-nary (p,prime) situations.

Green and Dimond (April,1970) have made attempts to illustrate that similar or equivalent operations exist for the nonlinear cyclic f.s.r., once a polynomial form is assumed for the feedback logic. They have tried to solve the problem in accordance with the following steps :

1. Two irreducible polynomials are considered.
2. The sequences associated with these polynomials are set out.
3. Using nonlinear multiplication procedure, the product polynomial is determined.
4. Considering the product polynomial independently, its cycle set is evaluated.
5. Finally, the cycle set of the product polynomial is compared with those of the factor polynomials and some general relationships between their cycle lengths are broughtforth.

However, the problem of relating the sequences of the product f.s.r to the sequences of its factors remained unsolved.

In fact, in concluding their paper on nonlinear product feedback shift registers (April, 1970) Green and Dimond have remarked that " A more difficult task in this field is the prediction of the cycle set of the composite f.s.r from a knowledge of the cycle sets of the factor f.s.r.s. ... Apart from the general relationships, the exact nature of the cycle set formation has not yet been revealed... "

This section provides a satisfactory solution to the above-mentioned problem. Specifically, a method is described here to predict the exact cycle set of the composite f.s.r from a knowledge of the factor f.s.r.s. As will be seen, the method uses only the basic principles concerning the sequential operation of an f.s.r. It is, thus, a remarkably simple and straight approach to solving the problem. It is shown to give valid results irrespective of whether the factor polynomials are repeated and / or nonrepeated. Specific illustrations are considered to cover the possible situations.

6.5.1 The sequential operation of a cascade set of f.s.r.s

Similar to the linear case, it is possible to reproduce the sequences of a composite nonlinear f.s.r by a sequential circuit involving a cascade of the factor f.s.r.s. This,of course, implies that at least one of the factor f.s.r.s is nonlinear.

In view of this, before considering the procedure for predicting the cycle set of the product f.s.r from the cycle-sets of the factor f.s.r.s, it is necessary to provide a more basic understanding of the sequential operation of a cascade set of f.s.r.s, in which at least one is nonlinear.

This section is meant for this purpose.

In Fig. (6.2a) are shown configurations of two f.s.r.s F_1 and F_2, whereby F_1 is linear and F_2 is nonlinear. Let $f_1(x)$ and $f_2(x)$ represent the characteristic polynomials of the units F_1 and F_2 respectively. In Fig. (6.2b), the units F_1 and F_2 are shown cascaded to form a composite f.s.r, F. For this arrangement the characteristic polynomial $F(x)$ of the composite f.s.r F is expressed in terms of $f_1(x)$ and $f_2(x)$ as :

$$F(x) = f_1(x) * f_2(x) \quad \ldots \quad \ldots \quad (6.21)$$

Suppose that -

F_1 consists of n stages,

F_2 consists of m stages,

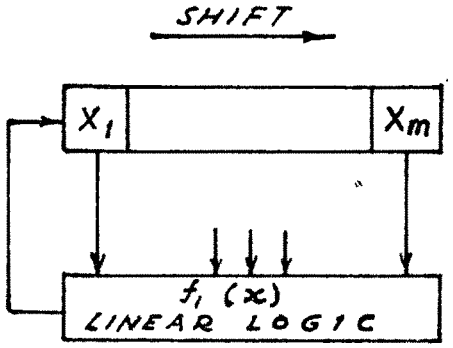
so that F consists of (m+n) stages.

In general, let

$\{a_{i,j}\}$ be the j sequences generated by F_1 with periods P_j

$\{b_{i,k}\}$ be the k sequences generated by F_2 with periods Q_k

$\{e_{i,r}\}$ be the r sequences generated by F with period R_r.

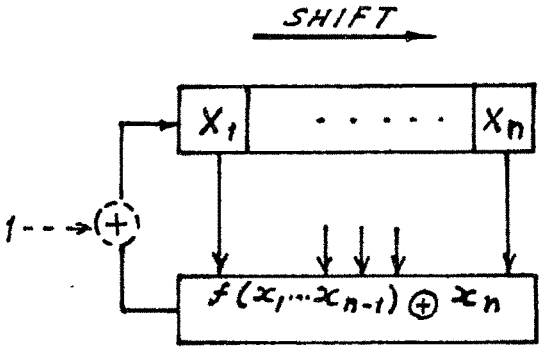


$$f_1(x) = \sum_{i=0}^{m} \oplus a_i \cdot x_n$$

WHERE $a_0 = a_m = 1$

AND $a_i = 0$ OR 1

F_1: LINEAR F.S.R. OF ORDER m



$$f_2(x) = f(x_1, x_2 \cdots, x_{n-1}) \oplus x_n$$

OR

$$f_2(x) = f(x_1, x_2 \cdots, x_{n-1}) \oplus x_n \oplus 1$$

(NONLINEAR TERM)

(SHOWN IN DOTTED LINE IN FIG.)

F_2: NON-LINEAR F.S.R. OF ORDER n

FIG. 6·2 (a)

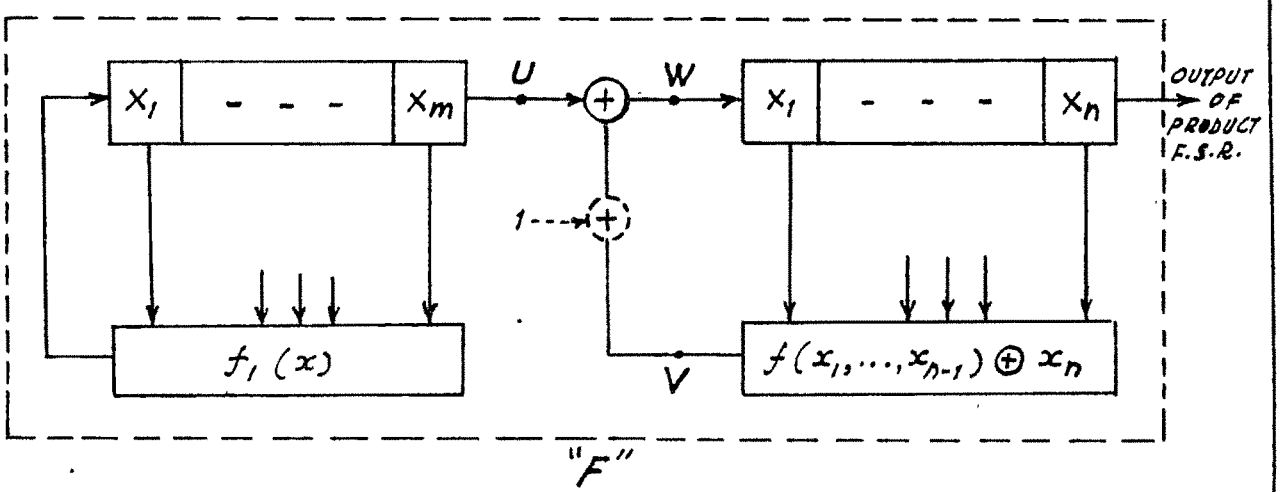


FIG. 6·2 (b) "F": NONLINEAR COMPOSITE OR PROCUCT F.S.R·OF ORDER $(m+n)$.

In particular, the nature of the cycle sets $\{a_{i,j}\}$, $\{b_{i,k}\}$, and $\{c_{i,r}\}$ depends on the factorable features of the polynomials $f_1(x)$ and $f_2(x)$ of the factor f.s.r.s. F_1 and F_2. However, the main feature of present interest is that :

the set $\{a_{i,j}\}$ describes all the 2^n possible states of F_1,

the set $\{b_{i,k}\}$ describes all the 2^m possible states of F_2,

and the set $\{c_{i,r}\}$ describes the 2^{m+n} possible states of F.

Now, from figure (6.2b), it is obvious that for a fixed initial state of the factor f.s.r F_1, the sequence generated at the output of the cascade - configuration depends on the initial state of the factor f.s.r F_2. Thus, with fixed initial state of F_1, when all the 2^m initial states of F_2 are considered, some members of the cycle set $\{c_{i,r}\}$ of the product f.s.r F will be obtained. Repeating this with all the remaining 2^n-1 initial states of F_1 leads to the remaining members of the set $\{c_{i,r}\}$.

Since, the present interest is only to examine the sequential operation of the cascaded set, let the initial states of F_1 and F_2 be fixed, and represented respectively by $(a_{-1}\ a_{-2} \cdots a_{-n})$ and $(b_{-1},\ b_{-2} \cdots b_{-m})$ so that :

F_1 generates the sequence $\{a_i\} = a_o, a_1, a_2, \cdots a_{P-1}$ of period P digits at the input of its first stage, and the product f.s.r F generates the sequence $\{c_i\} = c_o, c_1, \cdots c_{R-1}$ of period R digits at the mth stage of F_2 in the cascade set, shown in Fig. (6.2b).

Consider, now, the sequences at the locations labelled 'U,V,W' in the cascaded configuration; the sequence at the location U is the n-th stage output of F_1, i.e. $\{a_{i-n}\}$. Denoting the sequence at the location V by $\{d_i\} = (d_o, d_1, ..)$, the sequence e_i at the location W is given by :

$$\{e_i\} = (e_o, e_1, ..) = \{a_{i-n}\} \oplus \{d_i\} \quad ... \quad ... \quad (6.22)$$

Further, from the Fig.(6.2b), the sequence $\{c_i\}$ of the composite f.s.r is merely the sequence $\{e_i\}$ delayed by m digits; i.e

$$\{c_i\} = \{e_{i-m}\} \quad ... \quad ... \quad ... \quad (6.23)$$

Thus, the sequence $\{c_i\}$ of the product f.s.r is governed by the sequence $\{a_{i-1}\}$ of the factor f.s.r F_1 and the sequence $\{d_i\}$ which represents the output of the feedback logic device of F_2 in the cascade-operation.

Considering eqn. (6.22), when $\{a_{i-n}\}$ is the 'all-zero' sequence, $\{e_i\}$ is the same as $\{d_i = b_i\}$ which is the autonomous or natural respohse of the factor f.s.r F_2. When $\{a_{i-n}\}$ is different from 'all-zeros', the sequential operation of the composite f.s.r may be presented as follows :

Initially, state of F_1 is : $(a_{-1}\ a_{-2} \cdots a_{-n})$ and

state of F_2 is : $(b_{-1}\ b_{-2} \cdots b_{-m})$.

After setting the above initial states, the logic function $f_1(x)$ computes the input digit 'a_o' to F_1, and the logic function $f_2(x)$ computes the digit 'd_o' at the location V in the Fig.6.2b). And from eqn. (6.22), $\{e_i\}$ starts with the digit $e_o = a_{-n} \oplus d_o$.

At the arrival of the first shift pulse , the state of F_1 is changed to : ($a_o\ a_{-1}\ a_{-2} \cdots a_{-n+1}$) and the state of F_2 is changed to : ($e_o\ b_{-1}\ b_{-2} \cdots b_{-m+1}$). After these new states established, $f_1(x)$ computes the input 'a_1' to F_1 and $f_2(x)$ computes the digit 'd_1' at the location V. Again, from eqn.(6.22), the sequence $\{e_i\}$ takes the next digit 'e_1' given by

$$e_1 = a_{-n+1} \oplus d_1$$

This chain of action takes place untill at the (m+1)th state F_2 assumes the state : ($e_{m-1}\ e_{m-2} \cdots e_o$). This is the starting state of the cycle c_i at the mth stage of F_2 as stated in eqn. (6.23). Thus the first m states are merely transient states and do not get repeated. In short, the cyclic action of the product f.s.r starts from (m+1) state. This implies that a cycle of $\{c_i\}$ of period R appears at the mth stage of F_2 by the end of (R+m)-th state and new cycle starts from (R + m +1)th state. This sequential operation is shown in Table 6.4.

As seen from the Table 6.4, it may be said that the sequential behaviour of the product f.s.r whose polynomial F(x) is the product $f_1(x) * f_2(x)$ can be related to the autonomous behaviour of $f_1(x)$ and the forced response of $f_2(x)$ to the natural response of $f_1(x)$.

Based on these ideas, a procedure is described in the following sub-section to predict the cycle sets of the composite f.s.r F from the knowledge of the cycle sets of the factor f.s.r.s F_1 and F_2.

Table 6.4 : Sequential operation of the product f.s.r F, as mechanized by a cascade of factor f.s.r F_1 into the factor f.s.r F_2 as shown in Fig.(6.2b)

State No.	Locations referred to in the Fig. (6.2b) U $\{a_{i-n}\}$	V $\{d_i\}$	W $\{e_i\} = \{a_{i-n}\} \oplus \{d_i\}$	Caseaded configuration : stages of f.s.r F_2 x_1	x_2	x_3	x_m
1	a_{-n}	d_o	e_o	b_{-1}	b_{-2}	b_{-3}	b_{-m}
2	a_{-n+1}	d_1	e_1	e_o	b_{-1}	b_{-2}	b_{-m+1}
3	a_{-n+2}	d_2	e_2	e_1	e_o	b_{-1}	b_{-m+2}
. . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
. . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
m	a_{m-n-1}	d_{m-1}	e_{m-1}	e_{m-2}	e_{m-3}	e_{m-4}	b_{-1}
Cyclic operation of the product f.s.r F begins at the (m+1)th state $e_{i-m} = c_i$							
(m+1)	a_{m-n}	d_m	e_m	e_{m-1}	e_{m-2}	e_{m-3}	e_o
(m+2)	a_{m-n+1}	d_{m+1}	e_{m+1}	e_m	e_{m-1}	e_{m-2}	e_1
. . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
. . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
(R+m)	$a_{R+m-n-1}$	d_{R+m-1}	e_{R+m-1}	e_{R+m-2}	e_{R+m-3}	e_{R+m-4}	e_{R-1}
(R+m+1)	a_{R+m-n}	d_{R+m}	e_{R+m}	e_{m-1}	e_{m-2}	e_{m-3}	e_o
	$=(a_{m-n})$	$=(d_m)$	$=(e_m)$				

6.5.2 Procedure for prediction of cycle set of the composite f.s.r from the cycle sets of its factor f.s.r.s.

The statement of the problem is :

Given two polynomials $f_1(x)$ and $f_2(x)$ of which at least one is nonlinear, it is desired to predict the cycle set of the product polynomial $F(x) = f_1(x) * f_2(x)$.

Let the cycle set of $f_1(x)$ be denoted by $\{a_{i,j}\}$, of $f_2(x)$ by $\{b_{i,k}\}$, and of $F(x)$ by $\{c_{i,r}\}$. The product f.s.r F described by $F(x) = f_1(x) * f_2(x)$ can be mechanized by a cascade of f.s.r F_1 described by $f_1(x)$ into the f.s.r F_2 described by $f_2(x)$ as shown previously in Fig. (6.2b). The sequential operation of the so formed product f.s.r, F, is presented in the previous section; and in accordance with the ideas exposed therein, a procedure is described here to predict the cycle set of the product f.s.r F from the knowledge of its factor f.s.r.s. F_1 and F_2.

From eqns. (6.22) and (6.23) of the previous section, it is clear that the problem of predicting the cycle set $\{c_{i,r}\}$ of the product f.s.r now settles down to one of predicting the sequence $\{d_i\}$ at the location V in the cascaded configuration of Fig. (6.2b).

As seen from the referred figure, the sequence $\{d_i\}$ is computed by the feedback logic function $f_2(x)$ for all possible consecutive states assumed by the factor f.s.r F_2 during the cascaded operation.

Now, the cycle set $\{b_{i,k}\}$ of factor f.s.r F_2 describes all the possible states of F_2. Hence, any state of F_2 in the cascaded operation must be described by a member of the set $\{b_{i,k}\}$. Therefore, identifying the state of F_2 under consideration with a member of the cycle set $\{b_{i,k}\}$, the digit next (right side) to the identified m-digit number in that member is the digit of $\{d_i\}$ corresponding to the considered state of f.s.r. F_2.

Once the digit of $\{d_i\}$ corresponding to a state of F_2 is known, then the corresponding digit of $\{e_i\}$ can be obtained from eqn. (6.22). Again considering the next state of F_2, the next digit in $\{d_i\}$ and hence in $\{e_i\}$ can be similarly obtained. Persuading this procedure till F_2 assumes the state $e_{m-1}, \ldots, e_o$ produces a sequence of the cycle set $\{c_{i,r}\}$ of the product f.s.r. at the mth stage of F_2 of Fig. (6.2b).

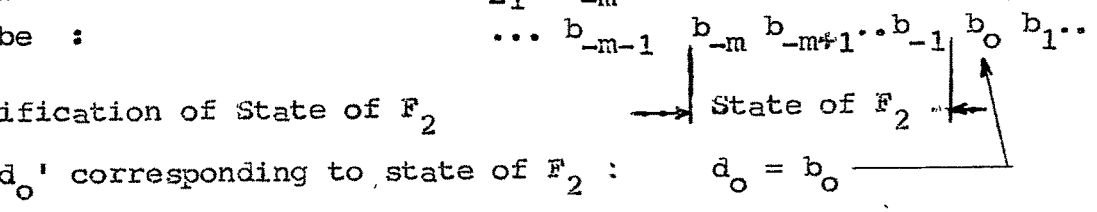
The procedure for predicting the entire cycle set $\{c_{i,r}\}$ of the product f.s.r F may be mechanized in accordance with the following steps :

Step 1 : Write the cycle sets $\{a_{i,j}\}$ and $\{b_{i,k}\}$ of the factor f.s.r.s. F_1 and F_2 that are characterized by the given polynomials $f_1(x)$ and $f_2(x)$ respectively.

Step 2A : Consider an initial state $(a_{-1} \ldots a_{-n})$ of F_1, and $(b_{-1} \ldots b_{-m})$ of F_2. Identify the state $(b_{-1} \ldots b_{-m})$ of F_2 with a member of the cycle set $b_{i,k}$ of F_2; and determine the digit d_o in the sequence $\{d_i\}$ corresponding to this state of F_2 as

shown below :

Let succession of digits in the member of $\{b_{i,k}\}$ which contains the state $(b_{-1}..b_{-m})$ of F_2 be :



Digit 'd_o' corresponding to state of F_2 : $d_o = b_o$

From eqn. (6.22), $\{e_i\}$ is obtained by mod-2 addition of $\{a_{i-n}\}$ and $\{d_i\}$. So, add the digit 'd_o' to 'a_{-n}' to obtain 'e_o'

Step 2B : At the arrival of the first shift pulse, the state of F_1 is : $(a_o\ a_{-1}...a_{-n+1})$; and the state of F_2 is : $(e_o\ b_{-1}..b_{-m+1})$. As in Step 2A, identify this state of F_2 with a member of $\{b_{i,k}\}$ and find the digit d_1. Add d_1 to a_{-n+1} to obtain e_1.

Step 2C: When the above procedure is continued till F_2 assumes the state $(e_{m-1}\ e_{m-2}..e_o)$, one sequence of the set $\{c_{i,r}\}$ of the composite f.s.r F is predicted, as shown below :

$\{a_{i-n}\}$	:	a_{-n}	a_{-n+1}	a_{-n+2}	.. a_{R+m-1}	a_{R+m}
$\{d_i\}$	:	d_o	d_1	d_2	.. d_{R+m-1}	d_{R+m}
$\{e_i\}$	:	e_o	e_1	e_2	... e_{R-1},	e_o

← one period →

Step 3: Keeping the initial state of F_1 unchanged, repeating the step 2 with the 2^m-1 remaining possible initial states of F_2 some sequences of the set $\{c_{i,r}\}$ are obtained. In this process, phase shifted versions of the same sequence do not add to the number of members of the set $\{c_{i,r}\}$.

Step 4 : So far, the initial state of F_1 has been kept unchanged. The factor f.s.r F_1 has n stages, and hence can have

in all 2^n possible initial states of which so far only one is considered in the steps 2 to 3. Now, repeating the steps 2 and 3 with the 2^n-1 remaining initial states of F_1 leads to forming the sequences of the set $\{c_{i,r}\}$ that have remained through the steps 2 and 3.

Step 5 : All the sequences formed through the steps 2 to 4 form the total cycle set $\{c_{i,r}\}$ of the product f.s.r F, which is characterized by the polynomial $F(x) = f_1(x) * f_2(x)$. To

To clear the concepts, an example is illustrated below :

Example 6.3 : Given two f.s.r.s. F_1 and F_2 characterized respectively by the polynomials $f_1(x)$ and $f_2(x)$ given by :

$x_o = f_1(x) = x_1 \oplus x_2$ (n = 2), linear polynomial

$x_o = f_2(x) = x_1 \oplus x_2 \oplus x_1 * x_2 \oplus x_4$ (m=4, nonlinear polynomial

It is desired to predict the cycle set of the product f.s.r. characterized by the logic function, $F(x) = f_1(x) * f_2(x)$.

The solution to the problem is given in accordance with the steps outlined earlier.

Step 1 : The cycle sets $\{a_{i,j}\}$ and $\{b_{i,k}\}$ of factor f.s.r.s F_1 and F_2 are seen to be :

$\{a_{i,1}\}$:	1 1 0	length 3
$\{a_{i,2}\}$:	0	length 1
$\{b_{i,1}\}$:	1 1 1 1 0 0 1 0 1 1 0 1 0 0 0	length 15
$\{b_{i,2}\}$:	0	length 1

Step 2A, 2B, 2C : F_1 has n=2 stages. Hence it has 2^2 possible initial states. F_2 has m=4 stages. Hence it has 2^4 possible initial states.

Let the initial (1st) state of F_1 be : $(a_{-1}a_{-2}) = (1\ 1)$

Let the initial (1st) state of F_2 be : $(b_{-1}b_{-2}b_{-3}b_{-4})$ (1 1 1 1)

The member of the set $\{b_{i,k}\}$ that describes the 1st state of F_2 : : $\{b_{i,1}\}$

Succession of digits in $b_{i,1}$: 1 1 1 1 0 0 1 0 1 1 0 1 0 0 0

Identification of 1st state of F_2 :

Digit d_o in d_i that corresponds to the 1st state of F_2 as given in Table (6.4) $d_o = 0$

From eqn. (6.22), $e_o = a_{-2} \oplus d_o$; hence $e_o = 1 \oplus 0 = 1$

Repeating this procedure till F_2 assumes the state : $(e_3e_2e_1e_o)$ gives a member, say $\{c_{i,1}\}$ of the cycle set $\{c_{i,r}\}$ of the product f.s.r. F as shown below :

$\{a_{i-n}\}$: 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0...

$\oplus\{d_i\}$: 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0...

$\{e_i\}$: 1 1 0 1 1 0 0 0 1 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 0...

1 cycle of length 24 digits

State of F_2 $b_4b_3b_2b_1e_oe_1e_2e_3$

1st

2nd

3rd

4th etc.

Digits in the sequence $\{d_i\}$, written above, are obtained from the set $\{b_{i,k}\}$ just in the same way as 'd_o' is obtained above. (shown below)

$\{b_{i,1}\}$:	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0
	d_8	d_9	d_{15}	d_{23}	d_0	d_3	d_7	d_{18}	d_{10}		d_5	d_6	d_4		d_{11}
	d_{12}	d_{14}	d_{19}		d_1		d_{17}				d_{22}	d_{16}	d_{21}		
					d_2							d_{20}			
$\{b_{i,2}\}$:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					d_{13}										

Step 3 : Keeping the initial state of F_1 unchanged, repeating the step 2 above with the remaining 2^4-1 initial states of F_2 gives some more sequences of the cycle set of product f.s.r F. The distinct sequences (not merely phase-shifted versions) so obtained are given in tabular form below.

Step 4 : F_1 has n=2 stages and so can have 2^2 possible initial states. So far, only the state (1,1) is considered. By considering the initial states (0,1) and (1,0), only phase shifted versions of those sequences already obtained in steps 2 and 3 will occur. This is because of the fact that F_1 is a maximum linear f.s.r. When F_1 is in all zero state, the product f.s.r is merely the factor f.s.r F_2 and so generates the sequences of F_2 already stated in step 1.

Step 5 : All the sequences of the product feedback shift register F governed by feedback logic function $F(x) = f_1(x) * f_2(x)$ are stated in tabular form below.

Initial state of F_1	Initial state of F_2	Members of the cycle set $\{c_{i,r}\}$ of the product f.s.r of order six	Length
(1 1)	(1 1 1 1)	110110001000011001101111	24
"	(1 1 0 0)	000101011100	12
"	(0 0 1 0)	010100111	9
(0 0)	(1 1 1 1)	111100101101000	15
"	(1 0 0 1)	100	3
"	(0 0 0 0)	0	1
		Length = 2^6	=64

Thus it is possible to predict the cycle set of the product feedback shift register from a knowledge of the cycle sets of its factors.

6.6 THE CONTRALINEAR f.s.r VIEWED AS A PRODUCT FEEDBACK SHIFT REGISTER

A simple contralinear f.s.r is linear apart from one inverter present in its feedback. A general contralinear circuit can be shown to be equivalent to a collection of simple contralinear circuits, each having at the most only one inverter (Bryant 1969). Hence, a contralinear f.s.r is formed by adding, in mod-2 sense, an all-1 sequence to the corresponding linear version. This 'all-1' sequence can be obtained by incorporating feedback into a 1-stage shift register and giving it an initial state '1'.

The resulting configuration involves a sequential circuit consisting of a cascade of 1-stage f.s.r into the first f.s.r with the logic function $F(x) = (1 \oplus x_1) * f(x)$. Such a circuit is either a cascade of $(1 \oplus x_1)$ into $f(x)$ or a single f.s.r with the polynomial $F(x)$. The cycles generated by such a circuit can be predicted viewing it as a product feedback shift register, as detailed in the previous section. The results obtained in an example are given below :

Example 6.4 : Given that the polynomials of the terms of a contralinear circuits when viewed as a cascaded set of $(1 \oplus x_1)$ into $f(x)$ as : $f(x) = x_o = x_2 \oplus x_3$, find the cycle set of $F(x) = (1 \oplus x_1) * f_1 x$

Let $f_1(x) = x_o = x_1$ represents the 1-stage f.s.r. Also, let $f_2(x) = x_o = x_2 \oplus x_3$ describe or stand for the given $f(x)$.

$f_1(x)$ generates two sequences :
- One sequence of all-ones, and
- one sequence of all-zeros.

$f_2(x)$ generates two-sequences :
- one sequence of all zeros and
- one sequence of period 7 digits : 1 1 1 0 0 1 0.

With $f_1(x)$ generating all zero sequence, product f.s.r is just the same as the linear part of the given contralinear f.s.r. But, when $f_1(x)$ generates all-1 sequence, assuming all the possible states of the linear term characterized by $f_2(x)$, the cycle set of the product f.s.r is determined by the method presented in the previous section, and the results so obtained are tabulated as under :

Initial state of 1-stage f.s.r	Initial state of the linear part of contralinear f.s.r	Cycle set of product f.s.r	Length (Length in digits)
(0)	(0 0 0)	0	1
(0)	(1 1 1)	1 1 1 0 0 1 0	7
(1)	(1 1 1)	1	1
(1)	(0 1 1)	1 0 0 0 1 1 0	7

6.7 SOME PROPERTIES OF NONLINEAR f.s.r SEQUENCES AND THEIR SEQUENCE DOMAIN CONSEQUENCES

From the analysis of nonlinear f.s.r.s so far carried out here, and what has been documented (See References in Sec. 6.1); some of the properties of nonlinear f.s.r.s and their sequences are listed in this section :

1. A nonlinear n-stage f.s.r generates closed branchless cycles when the nonlinear terms in the feedback logic are confined to the n-1 least significant variables, and the variable x_n occurs only as a linear sum, i.e. $F(x_1,..x_n)=f(x_1,..x_{n-1})\oplus x_n$. Absence of product terms in the logic leads to a linear f.s.r. Presence of x_n in any nonlinear term makes f.s.r noncyclic.
2. For an n-stage register, the total No. of f.s.r.s is 2^{2^n}. Of these, no. of f.s.r.s that produce pure cycles is $2^{2^{n-1}}$, which includes 2^{n-1} linear - and 2^{n-1} contralinear f.s.r.s.
3. For each function F(x) stated in property 1, there are 3 related forms: (1) The reverse function RF(x) describing time inverse sequences of F(x), (2) the dual function DF(x) having same cycles as of F(x) with 0 and 1 in them interchanged, and reverse dual function RDF(x) with cycles that are inverted cycles of RF(x)

4. The reverse of a composite polynomial is equal to the product of the reverses of its factors.
5. The dual of the product of two cyclic polynomials equals the product of the first with the dual of the second.
6. A cyclic polynomial and its dual are one and the same when $1 \oplus x_1$ is its factor.
7. Irreducible polynomials only generate odd number of cycles. Thus product polynomials generate only even number of cycles.
8. For F(x) to be skew-symmetric, condition is $F(x)=P(x).(1 \oplus x_1)$ and P(x) should generate cycles with odd number of 1s. Alternatively a cyclic polynomial is equal to its own dual if the polynomial is divisible by $1 \oplus x_1$, the division being consistent with post-multiplication by $1 + x_1$.
9. If RF(x) = F(x), i.e. when RDF(x) = DF(x), F(x) is reverse symmetric because each sequence of F(x) is accompanied by its reverse.

6.8 SUMMARY

A simple method is described in this chapter to form the polynomial version of a nonlinear logic, starting from the cycle structure of f.s.r The method is also useful to evaluate the cycle structure of a given logic. Further, a technique is introduced here for the prediction of the cycles of a product feedback shift register from a knowledge of its factors, thereby solving somewhat difficult and annoying problem in this nonlinear field. Also, the contralinear feedback shift registers are seen to be only a special class of nonlinear product feedback shift registers. Finally, it is seen that the classification of the f.s.r types has arisen by the definition of the reverse, and dual of a cyclic feedback function.

The polynomial form of feedback logic is thus seen to be quite a flexible theoretical description for the nonlinear case, and is likely to open up as yet abundant untapped source of codes for a number of new applications.
