JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 14, 645-667 (1998)

A General Structure of Feedback Shift Registers for Built-In Self Test

Kuen-Jong Lee, Wei-Lun Wang and Jhing-Fa Wang
Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan 701, R.O.C.
E-mail: kjlee@mail.ncku.edu.tw

A mixed-type feedback shift register (MFSR) is similar to a linear feedback shift register (LFSR) except that the connection between two consecutive flip-flops (F/F's) may be through the Q or output, and an extra inverter may exist at the input to the first flip-flop (stage) of the register. In this paper, we exploit the properties of MFSR's and show that by using an MFSR based pseudorandom pattern generator (PRPG) or multiple input signature analyzer (MISA), several good features for built-in self test can be obtained. Specifically we show that: (1) for any given initial seed, an MFSR always exists that can generate the same serial output sequence as can an LFSR with the same characteristic polynomial and any initial seed; (2) for any MFSR, we can always find an initial seed for this MFSR such that it can generate the same serial output sequence as can an LFSR with the same characteristic polynomial and any initial seed; and (3) for any given initial seed and any test response to be compressed, an MFSR based MISA can usually be found that will result in any required final signature. If such an MFSR cannot be found for a specific initial seed and a specific test response sequence, we show that by simply adding one arbitrary dummy pattern to the test response, one can always find the required MFSR. We also show that if the characteristic polynomial of the MFSR based MISA can be chosen freely, it is almost guaranteed that a feasible MFSR can be found without adding any dummy patterns to the test response. For example, for a 16-stage MISA, the probability that a feasible MFSR does not exist is less than 2-32768.

Keywords: mixed-type feedback shift register, linear feedback shift register, pseudorandom pattern generator, multiple input signature analyzer.

1. INTRODUCTION

Due to their simple and regular structure, linear feedback shift registers (LFSR's) and their extended versions have been widely used in the testing of digital circuits [1]. These include BILBO [2], M-LFSR [3], combined LFSR/SR [4], combined LFSR/XOR [5], cyclic LFSR [6], the LFSR based parallel pseudorandom pattern generator [7], the LFSR based pseudo-exhaustive test pattern generator [8, 9, 19], the reseeding and characteristic polynomial reprogramming techniques [10, 11], the multiple seed LFSR [12], the two-pattern generator [20] etc.

Received November 19, 1996; revised July 16, 1997.
Communicated by Youn-Long Lin.

The properties and theories of LFSR's have been discussed in [13, 14, 15, 16].

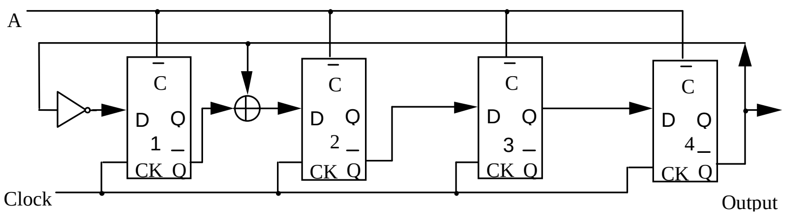
The applications of LFSR's to digital testing are found in two major fields. One is the use of an LFSR as a test pattern generator, known as a pseudorandom pattern generator (PRPG), and the other is the use of an LFSR as a test response compressor, known as a signature analyzer (SA) [1].

It is well known that an LFSR with a primitive characteristic polynomial can generate a maximum sequence (m-sequence) of length 2^k-1 if it contains a nonzero initial seed, where k is the number of F/F's in the LFSR [15]. Therefore, provision of a nonzero initial seed is an essential requirement for using an LFSR as a PRPG unless some extra logic is used. Conventionally, one may use F/F's with both preset and clear (reset) control lines, or use an externally controllable scan path to provide the nonzero seed. Both methods require extra area overhead [17].

For the signature analysis application, it is also clear that when the initial seed and test response are fixed, the final signature is fixed [15]. To determine the correctness of this final signature, either some storage for the correct signature and some comparison circuitry must be added to the circuit (chip) under test (CUT), or the signature must be *scanned* out of the CUT for comparison. Again, both methods require extra hardware overhead.

In this paper, we propose a generalized type of feedback shift registers, called a mixed-type feedback shift registers (MFSR's), that can be used to reduce the difficulty of the above problems. An MFSR is similar to an LFSR except that the connection between two consecutive stages of F/F's in an MFSR may be through the true (Q) or complementary ($\overline{Q}$) output of the preceding F/F stage, and the input to the first F/F stage may or may not go through an extra inverter. Fig. 1 shows an example of a 4-stage MFSR in which the output of F/F3 is through Q while F/F1, F/F2, and F/F4 are through $\overline{Q}$. An inverter presents before F/F1. Since, in general, both Q and $\overline{Q}$ of an F/F can be used for interstage connection, the circuit complexity of an MFSR is the same as that of an LFSR except that one extra inverter may be required at the input of F/F1.

The structure of an MFSR implies that an LFSR is a special case of an MFSR, where all the connections are through Q and no inverter presents before the first F/F stage. The intrainverted feedback shift register (IFSR) proposed in [17] is also a special case of an MFSR, where all the connections are through $\overline{Q}$, including the feedback from the last stage to the first stage. Since both LFSR's and IFSR's are special cases of MFSR's, all useful properties existing in LFSR's and IFSR's also exist in MFSR's. However, there are several unique properties of MFSR's that may not exist in LFSR's and IFSR's.



Clock	$\bar{Q}$	$\bar{Q}$	Q	$\bar{Q}$
0	1	1	0	1
1	1	1	1	1
2	1	1	1	0
3	0	0	1	0
4	0	1	0	0
5	0	1	1	1
6	1	0	1	0
7	0	0	0	0

Clock	$\bar{Q}$	$\bar{Q}$	Q	$\bar{Q}$
8	0	1	0	1
9	1	0	1	1
10	1	1	0	0
11	0	0	1	1
12	1	0	0	0
13	0	0	0	1
14	1	0	0	1
15	1	1	0	1

Fig. 1. Example of internal type MFSR with characteristic polynomial C(X) = X^4+X+1, initial seed (0000), and inversion vector (11101)

In this paper, we shall exploit these properties of MFSR's, emphasizing their application to random number generation and signature analysis. We shall show that: (1) for any given initial seed, an MFSR always exists that can generate the same output sequence as can an LFSR with the same characteristic polynomial and any initial seed; (2) for any MFSR, we can always find an initial seed for this MFSR such that it can generate the same output sequence as can an LFSR with the same characteristic polynomial and any initial seed; and (3) for any given initial seed and any test response to be compressed, an MFSR based MISA can usually be found that will result in any required final signature. If such an MFSR cannot be found for a specific initial seed and a specific test response sequence, we will show that by simply adding one arbitrary dummy pattern to the test response, one can always find the required MFSR. We will also show that if the characteristic polynomial of the MFSR based MISA can be chosen freely, it is almost guaranteed that a feasible MFSR can be found without adding any dummy pattern to the test response. For example, in a 16-stage MFSR based MISA, the probability that one would need an arbitrary dummy pattern is less than 2^{-32768}.

According to the above properties (1) and (2), we can see that the nonzero initial seed requirement for an LFSR based PRPG does not exist for an MFSR based PRPG. Hence, one can simply design an MFSR with the reset capability, which should be simpler than an LFSR with both the preset and reset capabilities. On the other hand, since the initial seed and the final signature of an MFSR based MISA can be chosen freely, much more design flexibility is allowed. For example, one may select an all-zero pattern for both the initial seed and the final signature, hence, simplifying the seeding and signature checking process. Another application is that if a register is to be used as an MISA in one test session and a PRPG in the next session, then we can set the final signature of the first session to be the same as the initial seed of the PRPG for the next session, hence, eliminating the need for seed reloading.

This paper is organized as follows. Section 2 gives a brief review of the well known LFSR's. Section 3 describes the structures and polynomial representation of MFSR's. Based on some formal theoretical analysis, Sections 4 and 5 relate the behaviors of the MFSR based PRPG and MISA to those of the LFSR based PRPG and MISA, respectively. A comparison between MFSR and IFSR based PRPG's is also given in Section 4. Finally, we give a discussion and draw conclusions in Section 6.

2. LINEAR FEEDBACK SHIFT REGISTERS

Depending on the positions of exclusive-or gates, there are two types of LFSR's. In this paper, we will only consider the internal type LFSR and its corresponding MFSR based on polynomial representation. The results obtained in this paper also apply to other types of LFSR's or MFSR's though analysis based on matrix representation may be necessary since it is difficult to conduct polynomial analysis on external type LFSR's or MFSR's. Hereafter, all LFSR's or MFSR's referred to are internal ones. The structure of a k-stage LFSR is shown in Fig. 2. The behavior of this LFSR can be described using mathematical polynomial representation over GF(2). Its characteristic polynomial C(X) is defined as $C(X) = \sum_{i=0}^{k} C_i * X^i$, where $C_0 = C_k = 1$. The symbols * and + denote binary multiplication and addition over GF(2), respectively. The initial state (a_{-1}, a_{-2},......., a_{-k}) can be represented using an initial state polynomial L0(X) as:

$$L_0(X) = \sum_{i=1}^{k} a_{-i} * X^{i-1}. \qquad \textbf{(1)}$$

The i-th state polynomial Li(X), i.e., the content of the k-stage LFSR after i shifts, is represented by

$$L_i(X) = [X^i * L_0(X) \bmod C(X)] \qquad i = 0, 1, 2, ... \qquad \textbf{(2)}$$

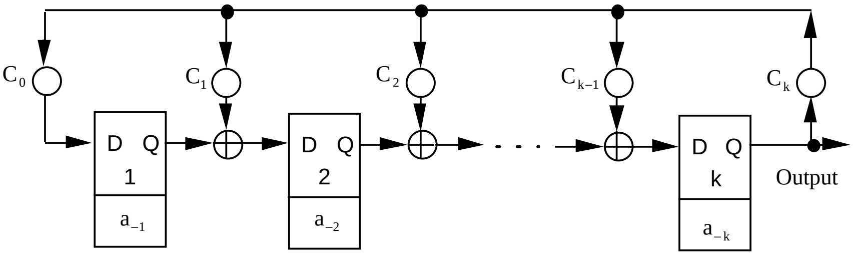
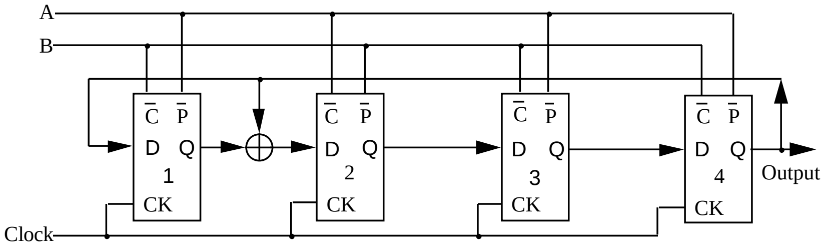


Figure 2 Architecture of internal type LFSR with characteristic polynomial $C(X) = \sum_{i=0}^{k} C_i * X^i$ and initial seed (a_{-1}, a_{-2}, ..., a_{-k})

This equation has also been used in [17], and it indicates that the i-th internal state in an LFSR is dependent on both the initial seed and the characteristic polynomial. Fig. 3 shows an example of an LFSR with $C(X) = X^4 + X + 1$ and an initial seed (0110), where and are clear and preset control lines to each F/F. When the initial state (1011) is needed, we let the values at A and B in Fig. 3 be 0 and 1, respectively.


A
B
$\overline{C}$ $\overline{P}$
D Q
1
CK
$\overline{C}$ $\overline{P}$
D Q
2
CK
$\overline{C}$ $\overline{P}$
D Q
3
CK
$\overline{C}$ $\overline{P}$
D Q
4
CK
Output
Clock

Clock	Q_1	Q_2	Q_3	Q_4
0	1	0	1	1
1	1	0	0	1
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	1	1	0	0
7	0	1	1	0

Clock	Q_1	Q_2	Q_3	Q_4
8	0	0	1	1
9	1	1	0	1
10	1	0	1	0
11	0	1	0	1
12	1	1	1	0
13	0	1	1	1
14	1	1	1	1
15	1	0	1	1

Figure 3 Example of internal type LFSR with characteristic polynomial $C(X) = X^4+X+1$ and initial seed (1011)

3. MIXED-TYPE FEEDBACK SHIFT REGISTERS

Refer to Fig. 4. The output of each F/F in a mixed-type feedback shift register (MFSR) is either in true (Q) or complementary ($\overline{Q}$) form and can be specified by the value of an inversion variable d_i. If the Q ($\overline{Q}$) output of the i-th F/F in MFSR is used, then d_i is zero (one). For the input to the first stage, which is fed by the output of the last stage, we assume that an inversion variable d_0 is used. Therefore, compared with an LFSR, the behavior of a k-stage MFSR can be characterized by a characteristic polynomial C(X) of degree k, an initial seed (s_{-1}, s_{-2},......, s_{-k}), and an inversion vector (d_0, d_1,......, d_k). It should be pointed out again that the Exclusive-OR gates and the d_i inputs shown in Fig. 4 are merely for our convenience in analyzing the properties of MFSR's. In the actual implementation, they need not exist. For example, Fig. 1 shows an MFSR with (d_0, d_1, d_2, d_3, d_4) = (1, 1, 1, 1, 0, 1).

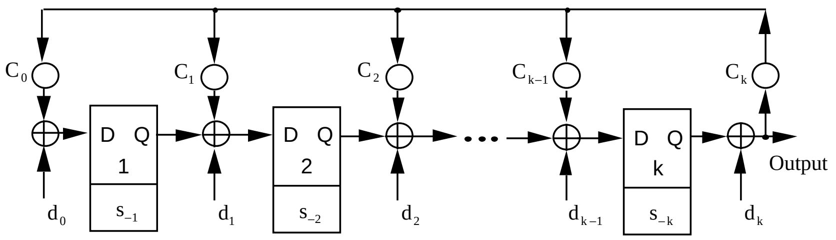


Figure 4 Architecture of internal type MFSR with characteristic polynomial $C(X) = \sum_{i=0}^{k} C_i * X^i$, initial seed $(s_{-1}, s_{-2}, ..., s_{-k})$, and inversion vector $(d_0, d_1, ..., d_k)$

Similar to the analysis of an LFSR, the initial state polynomial $T_0(X)$ of a k-stage internal type MFSR can be expressed as

$$T_0(X) = \sum_{i=1}^{k} s_{-i} * X^{i-1}. \tag{3}$$

Let $M_i(X)$ be a polynomial whose coefficients are the content of each F/F adding the corresponding inversion variable after i shifts and let the i-th internal state polynomial of the MFSR be $T_i(X)$ then $T_i(X)$ can be represented by

$$T_i(X) = M_i(X) + D(X) \tag{4}$$

where D(X) is the inversion polynomial of an MFSR and is defined as

$$D(X) = \sum_{i=1}^{k} d_i * X^{i-1}. \tag{5}$$

Note that $M_0(X)$ does not depend on d0 and can be expressed as:

$$M_0(X) = T_0(X) + D(X). \tag{6}$$

By the shift operation of an MFSR, we see that

$$\begin{aligned} M_1(X) &= [M_1(X) * X + D(X) + d_0] \bmod C(X) \\ &= [T_0 (X) * X + D(X) * (X+1) + d_0] \bmod C(X). \end{aligned}$$

$$\begin{aligned} M_2(X) &= [M_1(X) * X + D(X) + d_0] \bmod C(X) \\ &= [T_0(X) * X_2 + D(X) * (X_2 + X + 1) + d_0 * (X+1)] \bmod C(X). \end{aligned}$$

:

Therefore, we can obtain a general expression for $M_i(X)$ of an internal type MFSR:

$$
\begin{aligned}
M_i(X) &= [T_0(X) * X^i + D(X)*(X^i + X^{i-1} + \ldots + X + 1) + \\
&\quad d_0*(X^{i-1} + X^{i-2} + \ldots + X + 1)] \bmod C(X) \\
&= [T_0(X) * X^i + D(X)*(X^{i+1} + 1)/(X+1) + \\
&\quad d_0*(X^i + 1)/(X+1)] \bmod C(X). \qquad \textbf{(7)}
\end{aligned}
$$

Substituting this equation into Eq. (4), we obtain

$$
T_i(X) = \{T_0(X) * X^i + [D(X) * X + d_0]*(X^i + 1)/(X+1)\} \bmod C(X).
$$

$$
i = 0, 1, 2, \ldots \qquad \textbf{(8)}
$$

Fig. 1 shows an example of an MFSR with $C(X) = X^4 + X + 1$ and an initial seed (0000). Here, the inversion vector $(d_0, d_1, d_2, d_3, d_4)$ is (1, 1, 1, 0, 1). Note that the initial seed can be obtained by simply setting A to 0.

4. MFSR BASED PSEUDORANDOM PATTERN GENERATORS

In this section, we will discuss an MFSR based PRPG and its relation to an LFSR based PRPG. Let $\{O_m\} = \{O_0, O_1, O_2, \ldots\}$ and $\{O'_m\} = \{O'_0, O'_1, O'_2, \ldots\}$ represent the output sequence generated by an LFSR based PRPG and an MFSR based PRPG, respectively, where O_0 (O'_0) is the first bit appearing at the output, O1 (O'_1) is the second, and so on.

Fig. 2 shows the configuration of an internal type LFSR based PRPG. Each bit of the output sequence from this PRPG can be expressed as:

$$
\begin{aligned}
&O_0 = a_{-k} \\
&O_1 = C_{k-1} * O_0 + a_{-k+1} \\
&\quad \vdots \\
&O_{k-1} = C_{k-1} * O_{k-2} + C_{k-2} * O_{k-3} + .. + C_1 * O_0 + a_{-1} \\
&O_k = C_{k-1} * O_{k-1} + C_{k-2} * O_{k-2} + \ldots + C_2 * O_2 + C_1 * O_1 + O_0 \\
&O_{k+1} = C_{k-1} * O_k + C_{k-2} * O_{k-1} + \ldots + C_2 * O_3 + C_1 * O_2 + O_1 \\
&\quad \vdots
\end{aligned}
$$

The above equations can be generalized as:

$$
O_p = \sum_{i=1}^{p} (C_{k-i} * O_{p-i}) + a_{-k+p} \qquad \text{for } p = 0, 1, 2, \ldots, (k-1) \qquad \textbf{(9a)}
$$

or

$$O_p = \sum_{i=1}^{k-1}(C_{k-i} * O_{p-i}) + O_{-k+p} \qquad \text{for } p = k, (k+1), ... \qquad \textbf{(9b)}$$

Refer to Fig. 4. Assume that the MFSR under consideration has the same characteristic polynomial C(X); then, each bit of the output sequence from the k-stage MFSR is represented as:

$$
\begin{aligned}
O'_0 &= s_{-k} + d_k \\
O'_1 &= C_{k-1} * O'_0 + s_{-k+1} + d_{k-1} + d_k \\
&\vdots \\
O'_{k-1} &= C_{k-1}*O'_{k-2} + C_{k-2}*O'_{k-3} + .. + C_1*O'_0 + s_{-1} + d_1 + d_2 + ... + d_{k-1} + d_k \\
O'_k &= C_{k-1}*O'_{k-1} + C_{k-2}*O'_{k-2} + ... + C_2*O'_2 + C_1*O'_1 + O'_0 + d_0 + d_1 + d_2 \\
&\quad + ... + d_{k-1} + d_k \\
O'_{k+1} &= C_{k-1}*O'_k + C_{k-2}*O'_{k-1} + ... + C_2*O'_3 + C_1*O'_2 + O'_1 + d_0 + d_1 + d_2 \\
&\quad + ... + d_{k-1} + d_k \\
&\vdots
\end{aligned}
$$

The general form of the above equations is expressed as:

$$O'_p = \sum_{i=1}^{p}(C_{k-i} * O'_{p-i}) + s_{-k+p} + \sum_{i=0}^{p} d_{k-i} \qquad \text{for } p = 0, 1, 2, ..., (k-1) \qquad \textbf{(10a)}$$

or

$$O'_p = \sum_{i=1}^{k-1}(C_{k-i} * O'_{p-i}) + O'_{-k+p} + \sum_{i=0}^{k} d_i \qquad \text{for } p = k, (k+1), ... \qquad \textbf{(10b)}$$

In the following, Lemmas 1 and 2 show the requirement that both LFSR and MFSR based PRPG's can generate the same output sequences.

Lemma 1: For internal type LFSR and MFSR based PRPG's with the same characteristic polynomial of degree k, if they generate the same output sequence with initial seeds of (a_{-1}, a_{-2}, ..., a_{-k}) and (s_{-1}, s_{-2}, ..., s_{-k}), respectively, then the relationship between these initial seeds can be expressed as

$$a_{-i} + s_{-i} = \sum_{j=i}^{k} d_j \qquad \text{for } i = 1, 2, ..., k. \qquad \textbf{(11)}$$

Proof: Since the two output sequences of LFSR and MFSR based PRPG's are the same, from Eqs. (9a) and (10a), we have:

$O_0 = O'_0$, i.e., $a_{-k} = s_{-k} + d_k$

$O_1 = O'_1$, i.e., $C_{k-1}*O_0 + a_{-k+1} = C_{k-1}*O'_0 + s_{-k+1} + d_{k-1} + d_k$, which implies $a_{-k+1} = s_{-k+1} + d_{k-1} + d_k$.

:

$O_{k-1} = O'_{k-1}$, i.e., $C_{k-1}*O_{k-2} + C_{k-2}*O_{k-3} + .. + C_1*O_0 + a_{-1} = C_{k-1}*O'_{k-2} + C_{k-2}*O'_{k-3} + .. + C_1*O'_0 + s_{-1} + d_1 + d_2 + ... + d_{k-1} + d_k$, which implies $a_{-1} = s_{-1} + d_1 + d_2 + ... + d_{k-1} + d_k$.

These can be generalized as follows:

$$a_{-k+p} + s_{-k+p} = \sum_{i=0}^{p} d_{k-i} \qquad \text{for } p = 0, 1, 2, ..., (k-1).$$

Or equivalently, we have $a_{-i} + s_{-i} = \sum_{j=i}^{k} d_j$, for i = 1, 2, ..., k. **Q.E.D.**

Lemma 2: For internal type LFSR and MFSR based PRPG's with the same characteristic polynomial of degree k, if they generate the same output sequence, then the weight of the inversion vector in an internal type MFSR based PRPG must be even, where the weight of the inversion vector is defined as the summation of all the inversion variables.

Proof: Since internal type LFSR and MFSR based PRPG's with the same characteristic polynomial generate the same output sequence, by Eqs. (9b) and (10b), the k-th bit output sequence is

$O_k = O'_k$, i.e., $C_{k-1}*O_{k-1} + C_{k-2}*O_{k-2} + ... + C_2*O_2 + C_1*O_1 + O_0 = C_{k-1}*O'_{k-1} + C_{k-2}*O'_{k-2} + ... + C_2*O'_2 + C_1*O'_1 + O'_0 + d_0 + d_1 + d_2 + ... + d_{k-1} + d_k$,

and the (k+1)-th bit output sequence is

$O_{k+1} = O'_{k+1}$, i.e., $C_{k-1}*O_k + C_{k-2}*O_{k-1} + ... + C_2*O_3 + C_1*O_2 + O_1 = C_{k-1}*O'_k + C_{k-2}*O'_{k-1} + ... + C_2*O'_3 + C_1*O'_2 + O'_1 + d_0 + d_1 + d_2 + ... + d_{k-1} + d_k$

:

All these equations result in $d_0 + d_1 + d_2 + ... + d_{k-1} + d_k = 0$, i.e.,

$$\sum_{i=0}^{k} d_i = 0. \qquad \textbf{(12)}$$

Q.E.D.

Lemma 3: For two internal type LFSR and MFSR based PRPG's with the same characteristic polynomial of degree k, if Eqs. (11) and (12) hold, then the sequences generated by the two PRPG's are the same.

Proof: By Eq. (11), we have $a_{-i} + s_{-i} = \sum_{j=i}^{k} d_j$, and this can becomes

$$a_{-k} + s_{-k} = d_k$$

$$a_{-k+1} + s_{-k+1} = d_k + d_{k-1}$$

$$\vdots$$

$$a_{-1} + s_{-1} = d_k + d_{k-1} + ... + d_1 + d_0$$

By Eq. (12) we obtain

$$0 = d_k + d_{k-1} + ... + d_1 + d_0$$

Substituting these equations into Eq. (10a), we get

$$O'_0 = s_{-k} + d_k = a_{-k} = O_0$$

$$O'_1 = C_{k-1}*O'_0 + s_{-k+1} + d_{k-1} + d_k = C_{k-1}*O_0 + a_{-k+1} = O_1$$

$$\vdots$$

$$O'_{k-1} = C_{k-1}*O'_{k-2} + C_{k-2}*O'_{k-3} + .. + C_1*O'_0 + s_{-1} + d_1 + d_2 + ... + d_{k-1} + d_k = C_{k-1}*O_{k-2} + C_{k-2}*O_{k-3} + .. + C_1*O_0 + a_{-1} = O_{k-1}$$

$$O'_k = C_{k-1}*O'_{k-1} + C_{k-2}*O'_{k-2} + .. + C_1*O'_1 + O'_0 + d_0 + d_1 + d_2 + ... + d_{k-1} + d_k = C_{k-1}*O_{k-1} + C_{k-2}*O_{k-2} + .. + C_1*O_1 + O_0 = O_k$$

$$O'_{k+1} = C_{k-1}*O'_k + C_{k-2}*O'_{k-1} + .. + C_1*O'_2 + O'_1 + d_0 + d_1 + d_2 + ... + d_{k-1} + d_k = C_{k-1}*O_k + C_{k-2}*O_{k-1} + .. + C_1*O_2 + O_1 = O_{k+1}$$

$$\vdots$$

By induction the lemma holds. **Q.E.D.**

Theorem 1: If an internal type LFSR and MFSR have the same characteristic polynomial, then their output sequences are the same if and only if Eqs. (11) and (12) hold.

Proof: Lemma 1 and 2 give the necessary condition while Lemma 3 gives the sufficient condition. **Q.E.D.**

Given the initial seeds of an LFSR and an MFSR, the following example can be used to show how the inversion vector (d_0, d_1, d_2, ..., d_k) can be found.

Example 1: If the LFSR shown in Fig. 3, which has the initial state (a_{-1}, a_{-2}, a_{-3}, a_{-4}) = (1, 0, 1, 1), and the MFSR shown in Fig. 1, which has the initial state (s_{-1}, s_{-2}, s_{-3}, s_{-4}) = (0, 0, 0, 0), generate the same output sequence, then by Eqs. (11) and (12), we can obtain five equations as follows:

obtain five equations as follows:

$$a_{-4} + s_{-4} = d_4 = 1 \qquad \textbf{(13a)}$$

$$a_{-3} + s_{-3} = d_3 + d_4 = 1 \qquad \textbf{(13b)}$$

$$a_{-2} + s_{-2} = d_2 + d_3 + d_4 = 0 \qquad \textbf{(13c)}$$

$$a_{-1} + s_{-1} = d_1 + d_2 + d_3 + d_4 = 1 \qquad \textbf{(13d)}$$

$$d_0 + d_1 + d_2 + d_3 + d_4 = 0 \qquad \textbf{(13e)}$$

To solve these equations, firstly, d4 can be found in Eq. (13a); then, substituting d4 into Eq. (13b), d3 can be found and so on. Finally, d0 can be found in Eq. (13e). Thus, the value of the inversion vector (d_0, d_1, d_2, ..., d_4) in the MFSR is (1, 1, 1, 0, 1). In general, since a new inversion variable can be obtained in each equation by means of the above procedure, and there are totally (k+1) equations for the (k+1) unknowns, the solution is unique and is quite easy to obtain.

Based on Theorem 1, we can make two observations: (1) If the initial seed of an LFSR is given, then for any given initial seed, we can find an MFSR that can generate the same output sequence as can the LFSR. (2) If the inversion vector of an MFSR and the initial seed of an LFSR are given, then we can find an initial seed for this MFSR to generate the same output sequence. An immediate application of observation 1 is that the nonzero initial seed requirement for an LFSR based PRPG is no longer needed when an MFSR based PRPG is used. In fact, not only the all-zero initial seed, but also *any initial seed* can be used. This property gives somehow surprising results when compared with the IFSR based PRPG proposed in [17]. As mentioned before, an IFSR is a special case of an MFSR with all connections between two stages being through $\overline{Q}$. Our results here apparently state that given any LFSR with any initial seed, we can always find an IFSR that can generate an m-sequence that the LFSR can generate. However, in [17] it was stated that there exist some LFSR's that do not have corresponding IFSR's. Through careful examination, we can find that the inverter before the first F/F plays the major role in this difference. Due to this inverter, the weight of the inversion variable can be kept even; hence, we can always find an initial seed for any MFSR to generate the required m-sequence while in the IFSR design, freedom in selecting d0 is not allowed, which results in the inability to generate the output sequence of some LFSR's using IFSR.

So far, we have discussed the properties of the output sequences generated by MFSR and LFSR based PRPG's. Now, we will consider the internal states of MFSR based PRPG's and then discuss parallel pattern generators.

The following lemma shows the initial seed relationship between an LFSR and an MFSR from the point of view of polynomials.

Lemma 4: For internal type LFSR and MFSR based PRPG's with the same characteristic polynomial, if the relationship between their initial seeds can be expressed by Eq. (11), then the relationship between their initial state polynomials can be represented by

$$L_0(X) + T_0(X) = [X*D(X) + d_0] / (1 + X). \qquad \textbf{(14)}$$

Proof: The initial state polynomial $L_0(X)$ of a k-stage internal type LFSR and $T_0(X)$ of an internal type MFSR are expressed as $L_0(X) = \sum_{i=1}^{k} a_{-i} * X^{i-1}$ and $T_0(X) = \sum_{i=1}^{k} s_{-i} * X^{i-1}$.

By Eq. (11) the relationship between these initial seeds of the LFSR and the MFSR is expressed as $a_{-i} + s_{-i} = \sum_{j=i}^{k} d_j$ for i = 1, 2, ..., k.

Then,

$$L_0(X) = \sum_{i=1}^{k}(s_{-i} + \sum_{j=i}^{k} d_j) * X^{i-1}$$

$$= \sum_{i=1}^{k} s_{-i} * X^{i-1} + \sum_{i=1}^{k}\sum_{j=i}^{k} d_j * X^{i-1}$$

$$= T_0(X) + \sum_{i=1}^{k}\sum_{j=0}^{i-1} d_j * X^{i-1} \qquad (\text{since } \sum_{j=0}^{k} d_j = 0)$$

$$= T_0(X) + d_0 + X*(d_0 + d_1) + X^2*(d_0 + d_1 + d_2) + ... + X^{k-1}*(d_0 + d_1 + ... + d_{k-1})$$

$$= T_0(X) + d_0*(1 + X^k)/(1+X) + d_1*X*(1 + X^{k-1})/(1+X) + d_2*X^2*(1 + X^{k-2}) /(1+X) + ... + d_{k-1}*X^{k-1}*(1 + X)/(1 +X)$$

$= T_0(X) + [X*D(X) + d_0]/(1 + X)$.

Thus, the lemma. **Q.E.D.**

Theorem 2: For internal type LFSR and MFSR based PRPG's with the same characteristic polynomial C(X), their i-th internal state polynomials, $L_i(X)$ and T_i (X), can be expressed as

$$L_i(X) + T_i(X) = [X*D(X) + d_0] / (1 + X) \quad \textbf{(15)}$$

if and only if they generate the same output sequence.

Proof: By Theorem 1 and Eq. (8), the i-th internal state polynomial Ti(X) of an internal type MFSR based PRPG is expressed as: $T_i (X) = \{T_0(X)*X^i + [D(X)*X + d_0]*(X^i + 1)/(X+1)]\} \bmod C(X)$.

By Eq. (14), $T_i(X)$ can be rewritten as

$$T_i(X) = \{L_0(X)*X^i + X^i*(X*D(X) + d_0)/(X + 1) + (X*D(X) + d_0)*(X^i + 1)/(X + 1)\} \bmod C(X)$$

$$= \{L_0(X)*X^i + (X*D(X) + d_0)/(X + 1)\} \bmod C(X).$$

$$= L_i(X) + (X*D(X) + d_0)/(X + 1).$$

Thus, the theorem. **Q.E.D.**

Lemma 4 and Theorem 2 show that the relationship between the internal states of LFSR and MFSR based PRPG's is invariant. This means that (1) the sequence from any F/F output of an LFSR is the same as the sequence from the corresponding F/F output of the corresponding MFSR, and (2) each state transition of an LFSR from one state $L_i(X)$ to another state $L_j(X)$ of the LFSR has a corresponding state transition from one state $T_i(X)$ to another state $T_i(X)$ in an MFSR. Thus, the behavior of an LFSR can be completely mapped onto an MFSR. If it is required that the 2^k-1 parallel patterns generated by k-stage LFSR and MFSR based PRPG's be the same, then Theorem 2 can be used to determine how the output of an MFSR to a circuit under test can be tapped, as described next.

Let the i-th true outputs of LFSR and MFSR based PRPG's be $(Q_L)_i$ and $(Q_M)_i$, respectively. Then, Eq. (15) becomes

$$\sum_{i=1}^{k}[(Q_L)_i + (Q_M)_i]*X^{i-1} = [X*(\sum_{i=1}^{k} d_i * X^{i-1}) + \sum_{i=1}^{k} d_i] / (1+X)$$

$$= (\sum_{i=1}^{k} d_i) + X(\sum_{i=2}^{k} d_i) + X^2(\sum_{i=3}^{k} d_i) + \ldots + X^{k-1}(\sum_{i=k}^{k} d_i) \qquad \textbf{(16)}$$

By equating both sides of Eq. (16), we can obtain

$$(Q_L)_i + (Q_M)_i = \sum_{j=i}^{k} d_j . \qquad \textbf{(17)}$$

Eq. (17) indicates that to obtain the same parallel patterns as obtained in LFSR based PRPG's, the i-th true (complement) output of MFSR based PRPG's must be used if $\sum_{j=i}^{k} d_j = 0$ ($\sum_{j=i}^{k} d_j = 1$). For example, to generate the same parallel patterns in Fig. 3, the parallel pattern generator outputs in Fig.1 must be $((\overline{Q_1}, Q_2, \overline{Q_3}, \overline{Q_4}))$ with the inversion vector (d_0, d_1, d_2, d_3, d_4) = (1, 1, 1, 0, 1).

The following theorem shows how many different internal type MFSR based PRPG's can generate the same output sequence as that it can be generated by an internal type LFSR based PRPG.

Theorem 3: For a given characteristic polynomial of degree k, if the initial seed can be arbitrarily selected, then there are 2^k MFSR based PRPG's which can generate the same output sequence as that generated by an LFSR based PRPG with a given initial seed.

Proof: To meet the condition that any MFSR based PRPG can generate the same output sequence as that for an LFSR based PRPG, Eqs. (11) and (12) for any MFSR based PRPG must hold, namely, $a_{-i} + s_{-i} = \sum_{j=i}^{k} d_j$, and $\sum_{i=0}^{k} d_i = 0$.

There are (k+1) coefficients in the inversion vector, i.e.,

$$(d_0, d_1, d_2, \ldots, d_{k-1}, d_k) = (\sum_{i=1}^{k} d_i, d_1, d_2, \ldots, d_{k-1}, d_k)$$

Therefore, there are 2^k different combinations in the inversion vector if s_{-i}, i = 1, 2,......, k can be arbitrarily selected. Thus, the theorem holds. **Q.E.D.**

5. MULTIPLE INPUT SIGNATURE ANALYZERS (MISA's)

In this section, we will discuss the behavior of both LFSR based and MFSR based MISA's. We assume that the length of the test response sequence to be compressed is L. For a k-stage MISA, let its p-th input be I_p. We can use the following L*k array to represent the test response sequence:

$$\begin{array}{lllll}
R_{0,L-1} & R_{1,L-1} & R_{2,L-1} & \cdots & R_{k-1,L-1} \\
R_{0,L-2} & R_{1,L-2} & R_{2,L-2} & \cdots & R_{k-1,L-2} \\
: & : & : & : & : \\
R_{0,1} & R_{1,1} & R_{2,1} & \cdots & R_{k-1,1} \\
R_{0,0} & R_{1,0} & R_{2,0} & \cdots & R_{k-1,0}
\end{array}$$

where $R_{p,q}$ means the bit of the q-th test response applied to I_p of the MISA.

5.1 LFSR Based MISA's

Consider the LFSR based MISA shown in Fig. 5 that has a characteristic polynomial C(X) and an initial seed (a_{-1}, a_{-2},......, a_{-k}). The initial seed can also be expressed in terms of the polynomial $L_0(X)$ as stated in Eq. (1). If we put the L*k test response array into this LFSR based MISA, its i-th content after i shifts, $L_{-i}(X)$, can be expressed as:

$$\begin{aligned}
L_1(X) &= [X*L_0(X) + R_{0,L-1} + R_{1,L-1}*X + ... + R_{k-1,L-1}*X^{k-1}] \bmod C(X) \\
&= [X*L_0(X) + \sum_{j=0}^{k-1} R_{j,L-1} * X^j] \bmod C(X) \\
L_2(X) &= [X*L_1(X) + R_{0,L-2} + R_{1,L-2}*X + ... + R_{k-1,L-2}*X^{k-1}] \bmod C(X) \\
&= [X^2 * L_0(X) + \sum_{j=0}^{k-1} R_{j,L-1} * X^{j+1} + \sum_{j=0}^{k-1} R_{j,L-2} * X^j] \bmod C(X)
\end{aligned}$$

:

$$L_L(X) = [X^L * L_0(X) + \sum_{i=1}^{L}(\sum_{j=0}^{k-1} R_{j,L-i} * X^{j+L-i})] \bmod C(X) \qquad \textbf{(18)}$$

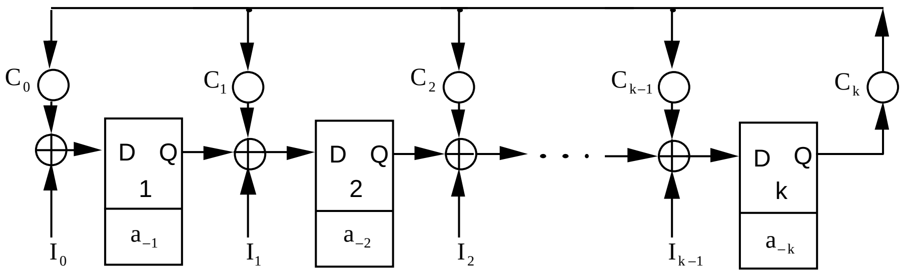


Figure 5 Architecture of internal type LFSR based MISA with characteristic polynomial $C(X)=\sum_{i=0}^{k} C_i * X^i$ and initial seed $(a_{-1}+a_{-2}+\ldots+a_{-k})$

LL(X) in Eq. (18) is the final signature of an internal type LFSR based MISA for an L*k test response array. Once the initial seed and characteristic polynomial have been specified, the final signature is determined by the given test response array.

5.2 MFSR Based MISA's

Fig. 6 shows a configuration of the MFSR based MISA. By means of the shift operation of an MFSR based MISA and an L*k test response array, we have

$$M_1(X) \quad = [X*M_0(X) + D(X) + d_0 + \sum_{j=0}^{k-1} R_{j,L-1} * X^j] \bmod C(X)$$

$$= [X*T_0(X) + D(X)*(X+1) + d_0 + \sum_{j=0}^{k-1} R_{j,L-1} * X^j] \bmod C(X)$$

$$M_2(X) \quad = [X*M_1(X) + D(X) + d_0 + \sum_{j=0}^{k-1} R_{j,L-2} * X^j] \bmod C(X)$$

$$= [X^2*T_0(X) + D(X)*(X^2+X+1) + d_0*(X+1) +$$

$$\sum_{j=0}^{k-1} R_{j,L-1} * X^{j+1} + \sum_{j=0}^{k-1} R_{j,L-2} * X^j] \bmod C(X)$$

$$\vdots$$

$$M_L(X) \quad = [X^L*T_0(X) + D(X)*(X^{L+1}+1)/(X+1) + d_0*(X^L+1)/(X+1) +$$

$$\sum_{i=1}^{L}\sum_{j=0}^{k-1} R_{j,L-i} * X^{L+j-i}] \bmod C(X). \qquad \textbf{(19)}$$

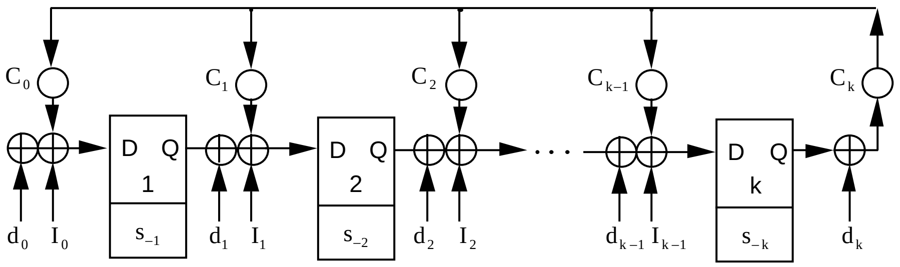


Figure 6 Architecture of internal type MFSR based MISA with characteristic polynomial $C(X)=\sum_{i=0}^{k} C_i * X^i$, initial seed $(s_{-1}, s_{-2}, ..., s_{-k})$, and inversion vector $(d_0, d_1, ..., d_k)$

ML(X) in Eq. (19) is the final signature of an internal type MFSR based MISA for an L*k test response array. Compared to Eq. (18), Eq. (19) indicates that even if the initial seed polynomial $T_0(X)$ and characteristic polynomial C(X) of an MFSR based MISA have been specified, it may still be possible that the final state polynomial $M_L(X)$ can be set to any desired value by adjusting the coefficients of the inversion vector (d_0, d_1,......, d_k). Therefore, MFSR based MISA's are also called seed-and-signature on demand (SASOD) MISA's. Next, we shall analyze this problem.

Let three polynomials, A(X), B(X), and F(X), be expressed as:

$$A(X) = M_L(X) + [X^L*T_0(X) + \sum_{i=1}^{L}\sum_{j=0}^{k-1} R_{j,L-i} * X^{L+j-i}] \bmod C(X) \qquad \textbf{(20)}$$

$$B(X) = (X^{L+1}+1)/(X+1) \qquad \textbf{(21)}$$

and

$$F(X) = (X^L+1)/(X+1). \qquad \textbf{(22)}$$

Substituting Eqs. (20), (21) and (22) into Eq. (19) we have

$$A(X) = [D(X)*B(X) + d_0*F(X)] \bmod C(X). \qquad \textbf{(23)}$$

The original meaning of Eq. (23) is that, given D(X), B(X), d_0*F(X), and C(X), it follows that A(X), which contains the final signature $M_L(X)$, is fixed. However, our objective here is to control the final signature to any required value by adjusting D(X) and d0. Therefore, our question becomes: Given any A(X), B(X), C(X), and F(X), can we find a D(X) and a d_0 such that Eq. (23) holds?

Since the behavior of feedback shift registers can be described by polynomial operations, we shall next solve this problem based on polynomial analysis. Let the greatest common divisor polynomial of B(X) and C(X) be G(X), i.e., GCD[B(X), C(X)] = G(X). We will first consider the case where $d_0 = 0$. In this case, we have

$$A(X) = [D(X)*B(X)] \bmod C(X). \qquad \textbf{(24)}$$

The following theorem gives the necessary and sufficient conditions for D(X) to exist.

Theorem 4: For any given three polynomials, A(X), B(X), and C(X), where deg A(X) < deg C(X). Eq. (24) has a solution on D(X) if and only if A(X) mod [GCD(B(X), C(X)] = 0.

Proof: It is well known that (see, e.g., [18]) G(X) = GCD[(B(X), C(X)] if and only if we can find two nonzero polynomials, u(X) and v(X), such that

$$G(X) = u(X)*B(X) + v(X)*C(X), \qquad \textbf{(25)}$$

and

$$GCD[u(X), v(X)] = 1. \qquad \textbf{(26)}$$

By means of the polynomial division algorithm, we can find a unique nonzero polynomial Q(X) such that Eq. (24) becomes

$$A(X) = D(X)*B(X) + Q(X)*C(X). \qquad \textbf{(27)}$$

(a) Sufficiency: Since A(X) mod G(X) = 0, we may assume that

$$A(X) = a(X)*G(X). \qquad \textbf{(28)}$$

By Eq. (24), we know that deg A(X) < deg C(X); hence, by Eqs. (28) and (25), we get

$$\begin{aligned} A(X) &= A(X) \bmod C(X) \\ &= [a(X)*G(X)] \bmod C(X) \\ &= [a(X)*u(X)*B(X) + a(X)*v(X)*C(X)] \bmod C(X) \\ &= [a(X)*u(X)*B(X)] \bmod C(X). \end{aligned}$$

By Eq. (5), if D(X) exists, then we know that deg D(X) < deg C(X). Compared to Eq. (27), we know that if we set

$D(X) = [a(X)*u(X)] \bmod C(X),$ **(29)**

then D(X) is a polynomial with degree less than deg C(X), and that D(X) satisfies Eq. (25). Therefore, a solution can be found.

(b) Necessity: Since $G(X) = GCD[B(X), C(X)]$,we can assume $B(X) = y(X)*G(X)$ and $C(X) = z(X)*G(X)$, where y(X) and z(X) are two nonzero polynomials.

Substituting them into Eq. (27), we get

$$A(X) = D(X)*y(X)*G(X) + Q(X)*z(X)*G(X)$$

$$= [D(X)*y(X) + Q(X)*z(X)]*G(X).$$

Thus, $A(X) \bmod G(X) = 0$. **Q.E.D.**

Now, we will consider the case where d0 = 1. In this case, we have

$A(X) = [D(X)*B(X) + F(X)] \bmod C(X).$ **(30)**

We modify the above equation and assume a polynomial A'(X) such that

$$A'(X) = A(X) + [F(X) \bmod C(X)].$$

Then, we have $A'(X) = [D(X)*B(X)] \bmod C(X).$ **(31)**

Since Eqs. (31) and (24) have the same form, we have the following theorem that gives the necessary and sufficient condition for D(X) to exist for $d_0 = 1$.

Theorem 5: For any given four polynomials, A(X), B(X), C(X), and F(X), where deg A(X) < deg C(X), Eq. (29) has a solution on D(X) if and only if A'(X) mod [GCD(B(X), C(X)] = 0, where A'(X) = A(X) + [F(X) mod C(X)].

Proof: Similar to Theorem 4. **Q.E.D.**

So far, we have proved the necessary and sufficient condition for D(X) to exist. Next, we will consider the case where d0 = 0 and A(X) mod [GCD(B(X), C(X))] * 0. The case where d0 = 1 can be similarly discussed. We have GCD[(XL+1+1)/(X+1), C(X)] * 1. Since, in general, C(X) is a primitive polynomial, in the following discussion, we shall make such an assumption. If GCD[(XL+1+1)/(X+1), C(X)] * 1, then GCD[(XL+1+1)/(X+1), C(X)] = C(X). The following theorem provides an approach to solve D(X) under this condition.

Theorem 6: If $GCD[(X^{L+1}+1)/(X+1), C(X)] \neq 1$ and C(X) is a primitive polynomial, then $GCD[(X^{L+2}+1)/(X+1), C(X)] = 1$.

Proof: Since (XL+2+1)/(X+1) + (XL+1+1)/(X+1) = XL+1, which cannot be a multiplication of C(X), the theorem is proved. **Q.E.D.**

Now, suppose we add one arbitrary dummy pattern to the test response such that $M_{L+1}(X)$ is the final signature of the MFSR based MISA for a (L+1)*k test response array, where

$$M_{L+1}(X) \quad = [X^{L+1}*T_0(X) + D(X)*(X^{L+2}+1)/(X+1) + d_0*(X^{L+1}+1)/(X+1) + \sum_{i=1}^{L+1}\sum_{j=0}^{k-1} R_{j,L+1-i} * X^{L+1+j-i}] \bmod C(X).$$

Then Eqs. (20) and (21) become

$$A_1(X) = M_{L+1}(X) + [X^{L+1}*T_0(X) + \sum_{i=1}^{L+1}\sum_{j=0}^{k-1} R_{j,L+1-i} * X^{L+1+j-i}] \bmod C(X)$$

and

$$B_1(X) = (X^{L+2}+1)/(X+1).$$

Since GCD[B_1(X), C(X)] = 1 if GCD[B(X), C(X)] ≠ 1, no matter what the value A_1(X) is, D(X) must always have a solution.

Now, we will describe a procedure to find D(X) if one exists. From Eq. (29), we know that if we can find u(X) and v(X) that satisfy Eqs. (25) and (26), then D(X) can be easily found. Next, we shall analyze the procedure for finding G(X) = GCD[B(X), C(X)]. From this procedure, we can derive a method to find u(X) and v(X). A standard method for finding the greatest common divisor of two polynomials B(X) and C(X) is to apply the Euclidean algorithm [21] successively as follows:

Euclidean algorithm [21] successively as follows:

$$B(X) = Q_1(X) * C(X) + R_1(X)$$

$$C(X) = Q_2(X) * R_1(X) + R_2(X)$$

$$R_1(X) = Q_3(X) * R_2(X) + R_3(X)$$

:

$$R_{k-3}(X) = Q_k(X) * R_{k-2}(X) + R_{k-1}(X)$$

$$R_{k-2}(X) = Q_{k+1}(X) * R_{k-1}(X) + R_k(X)$$

$$R_{k-1}(X) = Q_{k+2}(X) * R_k(X) + R_{k+1}(X)$$

until $R_{k+1}(X) = 0$. Then GCD[B(X), C(X)] = $R_k(X)$.

By reversing the above procedure, we can represent R_k (X) as a function of $R_{k-1}(X)$ and $R_{k-2}(X)$, then represent $R_{k-1}(X)$ as a function of $R_{k-2}(X)$ and $R_{k-3}(X)$, and so on. Finally, we can represent $R_k(X)$ as a function of B(X) and C(X) in the form $R_k(X) = u(X)*B(X) + v(X)*C(X)$. u(X) in this equation can then be used to determine D(X). We will use an example to illustrate the above process: let $A(X) = X^3 + X^2 + 1$, $B(X) = (X^{11} + 1)/(X + 1)$, and $C(X) = X^4 + X^3 + 1$. We have

$$B(X) = (X^6 + X^4 + X + 1) * C(X) + X^2$$

$$C(X) = (X^2 + X) * (X^2) + 1$$

$$X^2 = (X^2) * 1 + 0$$

Therefore,

$$\begin{aligned} \text{GCD}[B(X), C(X)] &= 1 \\ &= C(X) + (X^2 + X) * (X^2) \\ &= C(X) + (X^2 + X) * [B(X) + (X^6 + X^4 + X + 1) * C(X)] \\ &= (X^2 + X) * B(X) + (X^8 + X^7 + X^6 + X^5 + X^3 + X + 1) * C(X). \end{aligned}$$

Thus $u(X) = (X^2 + X)$. Since GCD[B(X), C(X)] = 1, a(X) = A(X). From Eqs. (28) and (29), we have

$$\begin{aligned} D(X) &= [a(X) * u(X)] \bmod C(X) \\ &= [(X^3 + X^2 + 1) * (X^2 + X)] \bmod (X^4 + X^3 + 1) \\ &= X^2 + 1. \end{aligned}$$

5.3 Consideration of the Dummy Pattern

We have shown that if D(X) does not exist for some specified initial seed and test response, then by simply adding one dummy pattern, one can always find the required D(X). In general, adding one dummy pattern should not be a problem because this only requires one more clock cycle during testing if a pseudorandom number generator is used for the pattern generator, or it requires that one more test pattern be scanned into the circuit under test if a scan system is used. In the case where such augmentation is absolutely not allowed, the following discussion shows that the problem may still be easy to solve.

From Theorem 5, we know that if {GCD[B(X), C(X)] = 1}, then a solution for D(X) always exists. Therefore, if we can use different C(X)'s, then the probability for {GCD[B(X), C(X)] $\neq$ 1} will be smaller. According to [15], the number of primitive polynomials with degree k is $(\phi(2^k-1))/k$, where the ϕ(X)-function is the Euler's function and is defined as the number of positive integers less than or equal to X that are relatively prime to X. If k = 16, then $(\phi(2^k-1))/k = 2048$. Now, if none of these primitive polynomials is prime to B(X), then the multiplication of these polynomials, denoted as MC(X), must also divide B(X). If the degree of MC(X) is N, then

$$N = k*(\phi(2^k-1))/k = \phi(2^k-1). \qquad \textbf{(32)}$$

The probability that MC(X) divides B(X) will be

$$P = 2^{\{\deg B(X)\}-N} / 2^{\{\deg B(X)\}} = 2^{-N}. \qquad \textbf{(33)}$$

If k = 16, then N = 16*2048 = 32768, and we have $P = 2^{-N} = 2^{-32768}$. Therefore, it is almost guaranteed that a feasible D(X) can be found if we have the freedom to choose C(X).

6. SUMMARY AND FUTURE WORK

In this paper, we have presented a generalized type of feedback shift register, called the mixed-type feedback shift register (MFSR),and shown that the conventional LFSR and the IFSR proposed in [17] are two special cases of MFSR's. Some properties that MFSR's possess while conventional LFSR's or IFSR's do not have been described. These properties are quite useful in built-in self-test. Compared to LFSR's, MFSR's do not require any extra hardware overhead except for the inverter at the input of the first stage when $d_0 = 1$.

Our analysis shows that by using MFSR based PRPG's, one can generate the same serial output sequence and parallel patterns as those generated by any LFSR based PRPG, with the extra advantage that the initial seed can be any value. On the other hand, the seed and signature of an MISA (or the final state at the true outputs of an MFSR) can be set to any value if an MFSR is used. One of the typical applications of this type of MISA is in mass production, where the same kind of CUT's are tested using BIST. In such a case, the initial seed and the final state can be set to the same value. If the final signature from a CUT is correct, then the seed need not be reloaded; thus, testing time can be reduced.

The analysis provided in this paper has been based on polynomial representation of feedback shift registers. It would be interesting to use matrix representa-tion to analyze MFSR's because the behaviors of external types of feedback shift registers are difficult to analyze using polynomial representation. With the introduction of MFSR's, we expect that many test schemes that have previously been developed using LFSR's can be reexamined to see whether better properties or architectures exist when MFSR's are used.

REFERENCES

1. M. Abramovici, M.A. Breuer and A.D. Friedman, *Digital Systems Testing and Testable Design,* Computer Science Press, New York, 1990.
2. B. Konemann, J. Mucha and G. Zwiehoff, "Built-in logic block observation technique," *Digest of Papers 1979 Test Conference*, 1979, pp. 37-41.
3. R. Raina and P.N. Marinos, "Signature analysis with modified linear feedback shift registers (M-LFSRs)," in *Proceedings of Fault-Tolerant Computing: 21st International Symposium*, 1991, pp. 88-95.
4. Z. Barzilai, D. Coppersmith and A.L. Rosenberg, "Exhaustive generation of bit patterns with applications to VLSI self-testing," *IEEE Transactions on Computers*, Vol. C-32, No. 2, 1983, pp. 190-194.
5. S.B. Akers, "On the use of linear sums in exhaustive testing," *Digest of Papers 15th Annual International Fault-Tolerant Computing Symposium*, 1985, pp. 148-153.
6. L.T. Wang and E.J. McCluskey, "Linear feedback shift register design using cyclic codes," *IEEE Transactions on Computers*, Vol. C-37, No. 10, 1987, pp. 1302-1306.
7. P.H. Bardell, "Design considerations for parallel pseudorandom pattern generators," *Journal of Electronic Testing and Applications*, Vol. 1, No. 1, 1990, pp. 73-87.
8. W.B. Jone and C.A. Papachristou, "A coordinated approach to partitioning and test pattern generation for pseudo-exhaustive testing," in *Proceedings of 26th ACM/IEEE Design Automation Conference*, 1989, pp. 525-530.
9. R. Srinivasan, S.K. Gupta and M.A. Breuer, "Novel test pattern generators for pseudo-exhaustive testing," in *Proceedings IEEE International Test Conference*, 1993, pp. 1041-1050.
10. S. Hellebrand, S. Tarnick, J. Rajski and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers," in *Proceedings of IEEE International Test Conference,* 1992, pp. 120-129.
11. S. Venkataraman, J. Rajski, S. Hellebrand and S. Tarnick, "An efficient BIST scheme based on reseeding of multiple polynomial linear feedback shift

registers," in *Proceedings of ICCAD-93*, 1993, pp. 572-577.

12. J. Savir and W.H. McAnney, "A multiple seed linear feedback shift register," *IEEE Transaction on Computers*, Vol. 41, No. 2, 1992, pp. 250-252.
13. R.A. Frohwerk, "Signature analysis: A new digital field service method," *Hewlett-Packard Journal*, Vol. 28, No. 9, 1977, pp. 2-8.
14. J.E. Smith, "Measures of the effectiveness of fault signature analysis," *IEEE Transactions on Computers*, Vol. C-29, No. 6, 1980, pp. 510-514.
15. P.H. Bardell, W.H. McAnney and J. Savir, *Built-in Test for VLSI: Pseudorandom Techniques*, Wiley, New York, 1987.
16. C.L. Chen, "Linear dependencies in linear feedback shift registers," *IEEE Transactions on Computers*, Vol. C-35, No. 12, 1986, pp. 1086-1088.
17. A. Guha and L.L. Kinney, "Relating the cyclic behavior of linear and intrainverted feedback shift registers," *IEEE Transactions on Computers*, Vol. C-41, No. 9, 1992, pp. 1088-1100.
18. R.E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, MA, 1983.
19. C.A. Chen and S.K. Gupta, "A methodology to design efficient BIST test pattern generators," in *Proceedings of IEEE International Test Conference*, 1995, pp. 814-823.
20. C.A. Chen and S.K. Gupta, "BIST test pattern generators for two-pattern testing - theory and design algorithms," *IEEE Transactions on Computers*, Vol. 45, No. 3, 1996, pp. 257-269.
21. I.N. Herstein, *Topics in Algebra*, 2nd (ed.), Wiley, 1975.

Kuen-Jong Lee（李昆忠）received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., the M.S. degree in electrical and com-puter engineering from the University of Iowa, Iowa City, Iowa, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, California. He joined the faculty of National Cheng-Kung University, Tainan, Taiwan, R.O.C., in 1991, and is currently a Professor in the Department of Electrical Engineering. His research interests include several aspects of computer-aided design and implementation of integrated circuits, with particular emphasis on automatic test pattern generation, design of testable circuits and design automation.

Wei-Lun Wang(王維倫)received the B.S. degree in electronic engineering from Chung Yuan Christian University in 1985, and the M.S. degree in electrical engineering from Tatung Institute of Technology in 1987. He is now a Ph.D. student in the Department of Electrical Engineering at National Cheng Kung University. His research interests are in the field of VLSI design and test.

Jhing-Fa Wang（王駿發）received the Ph.D. degree in electrical engineering and computer science from the Stevens Institute of Technology, Hoboken, in 1983. He is a senior member of IEEE and was elected general chairman of the Chinese Image Processing and Pattern Recognition Society in 1993. He was the director of the Institute of Information Engineering in National Cheng Kung University from 1990 to 1996. He is currently a professor in the department of Electrical Engineering and the Institute of Information Engineering at National Cheng Kung University. He is currently also the Chairman of the Taiwan Information Software Association and the Chairman of the Computer Center of National Cheng Kung University. His current research interests include graph theory, CAD/VLSI, neural nets for image processing, computer speech processing, and optical character recognition.