Technical Report

On Designing Transformed Linear Feedback Shift Registers with Minimum Hardware Cost

Laung-Terng Wang, Nur A. Touba, Richard P. Brent, Hui Xu, and Hui Wang

UT-CERC-12-03

November 8, 2011

Computer Engineering Research Center
Department of Electrical & Computer Engineering
The University of Texas at Austin

1 University Station, C8800
Austin, Texas 78712-0323
Telephone: 512-471-8000
Fax: 512-471-8967
http://www.cerc.utexas.edu

On Designing Transformed Linear Feedback Shift Registers with Minimum Hardware Cost

Laung-Terng Wang[1], Nur A. Touba[2], Richard P. Brent[3], Hui Xu[4], and Hui Wang[4]

[1] SynTest Technologies, 505 S. Pastoria Ave., Suite 101, Sunnyvale, CA 94086, USA
[2] Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712, USA
[3] Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia
[4] School of Microelectronics, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract

This paper provides a proof that given a standard or modular linear feedback shift register (LFSR) that uses k 2-input XOR gates to generate pseudorandom sequences, any transformed LFSR (t-LFSR) implementing the same characteristic polynomial, f(x), as the standard or modular LFSR cannot use fewer than $\log_2(k+1)$ *2-input XOR gates when k is an odd number, or* $1+\log_2 k$ *2-input XOR gates when k is an even number. This property applies to any n-stage t-LFSR design regardless of whether f(x) is a primitive polynomial or not. A new class of minimum-cost LFSRs (min-LFSRs) is subsequently developed to reduce the hardware cost to a minimum.*

1. Introduction

For decades, due to its simple circuit structure that consists of only flip-flops and a few 2-input XOR gates, *linear feedback shift registers* (LFSRs) have been widely used in the communication and computer industries to generate pseudorandom sequences. Applications of LFSRs include error correcting codes [1], pseudorandom pattern generation and signature analysis in logic *built-in self-test* (BIST) [2, 3], test data decompression and test data compaction in scan compression [3, 4], and cryptography [5].

Such LFSRs are typically constructed in a standard or modular form, where one or more XOR gates are interspersed between a flip-flop and the feedback path to generate a desired pseudorandom sequence [6]. When a maximum-length sequence (often called an ***m*-sequence**) is generated, the LFSR is referred to as a **maximum-length LFSR**. If k 2-input XOR gates are required to generate a pseudorandom sequence, then the signal on the feedback path would have to propagate through k XOR gates (as in the **standard LFSR**) or must be strong enough to drive k+1 fanout nodes (as in the **modular LFSR**). In either case, the circuit is slowed and may not be applicable for high-performance applications.

To improve the performance of these conventional LFSRs, many approaches have been proposed. Most noticeable are the solutions that include **decimations** that allow summing up several *m*-sequences produced by independent devices with a multiphase clock generator [7]; **windmill machines** that elevate a state transition rate but need additional registers [8]; **hybrid LFSRs** that reduce the number of XOR gates to $(k+1)/2$ when the characteristic polynomial, $f(x)$, generating an *m*-sequence meets certain requirement [9]; **ring generators** that enable each flip-flop output to drive at most 2 fanout nodes and introduce at most one level of one 2-input XOR gate between any two flip-flops, if its characteristic polynomial does not contain consecutive terms [10]; and **hybrid ring generators** that use the same number of XOR gates as their corresponding hybrid LFSRs [11] and preserve the high speed and simplified layout benefits of the ring generators, when the same requirement as the hybrid LFSRs is met.

While the high-performance and hardware cost issues have been respectively addressed in the literature, it is unclear in the design of hybrid LFSRs and hybrid ring generators whether a minimum hardware cost (in terms of the number of 2-input XOR gates required to construct the design) has been achieved. This paper is intended to answer this question. Based on the **transformation** properties given in [12], we will first illustrate by examples that a ***transformed LFSR*** (t-LFSR) implementing the same characteristic polynomial, $f(x)$, as a standard or modular LFSR that uses k 2-input XOR gates can use as low as $\log_2(k+1)$ XOR gates when k is an odd number, or $1+\log_2 k$ XOR gates when k is an even number, regardless of whether $f(x)$ is a primitive polynomial or not. We will then prove that given a standard or modular LFSR that uses k 2-input XOR gates to generate pseudorandom sequences, any t-LFSR implementing the same $f(x)$ as the standard or modular LFSR cannot use fewer than $\log_2(k+1)$ or $1+\log_2 k$ 2-input XOR gates, depending on odd or even k. The t-LFSR design that uses a minimum number of 2-input XOR gates is referred to a ***minimum-cost LFSR*** (*min*-LFSR).

This paper shows that it is possible to construct a t-LFSR that uses a fewer number of XOR gates than its hybrid LFSR or hybrid ring generator counterpart. However, the t-LFSR design that leads to a *min*-LFSR may lose the highly regular or modular structure which is a major benefit of using the (hybrid) ring generator design. A quick visual inspection rule of thumb and a simple construction method are given so one needs not to go through the complex transformations to avoid errors.

2. Background

There are two conventional forms of LFSR designs: standard LFSR and modular LFSR. Despite different state trajectories, both structures are capable of generating an *m*-sequence for each stage output.

2.1 Standard LFSRs

Fig. 1 shows an *n*-stage standard LFSR. It consists of *n* flip-flops and a number of XOR gates. Since XOR gates are placed on the external feedback path, the standard LFSR is also referred to as an **external-XOR LFSR** [6].

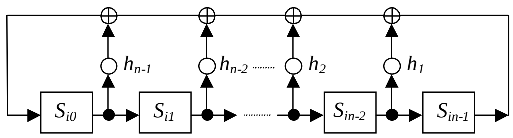


Figure 1. An *n*-stage (external-XOR) standard LFSR.

2.2 Modular LFSRs

Similarly, an *n*-stage modular LFSR with each XOR gate placed between two adjacent flip-flops, as shown in Fig. 2, is referred to as an **internal-XOR LFSR** [6]. This circuit runs faster than its corresponding standard LFSR, because each stage introduces at most one XOR-gate delay.

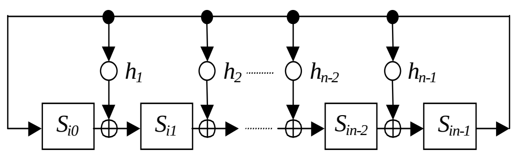


Figure 2. An *n*-stage (internal-XOR) modular LFSR.

2.3 LFSR Properties

The internal structure of the *n*-stage LFSR in each figure can be described by specifying a **characteristic polynomial** of degree *n*, $f(x)$, in which the symbol h_i is either 1 or 0, depending on the existence or absence of the feedback path, where

$$f(x) = 1 + h_1x + h_2x^2 + \ldots + h_{n-1}x^{n-1} + x^n. \quad (1)$$

Let S_i represent the contents of the *n*-stage LFSR after *i*th shifts of the initial contents, S_0, of the LFSR, and $S_i(x)$ be the polynomial representation of S_i, where $i \geq 0$. Then, $S_i(x)$ is a polynomial of degree n–1, where

$$S_i(x) = x^iS_0(x) \bmod f(x)$$
$$= S_{i0} + S_{i1}x + S_{i2}x^2 + \ldots + S_{in-2}x^{n-2} + S_{in-1}x^{n-1}. \quad (2)$$

If T is the smallest positive integer such that $f(x)$ divides $1+ x^T$, then the integer T is called the **period** of the LFSR. If $T = 2^n – 1$, then the *n*-stage LFSR generating the maximum-length sequence or *m*-sequence is called a **maximum-length LFSR** and thus can serve as an MLSG.

Define a **primitive polynomial** of degree *n* over **Galois field** GF(2), $p(x)$, as a polynomial that divides $1 + x^T$, but not $1 + x^i$, for any integer $i < T$, where $T = 2^n–1$ [6]. A primitive polynomial is **irreducible**. For illustration purpose, Figs. 3 and 4 show a 5-stage standard LFSR and a 5-stage modular LFSR with $f(x) = 1 + x^2 + x^3 + x^4 + x^5$, respectively. As can be seen, each circuit uses a total of 3 2-input XOR gates. The output signal at flip-flop 4 needs to propagate through 3 XOR gates to reach flip-flop 0 in Fig. 3 or must be strong enough to drive 4 fanout nodes in Fig. 4. The characteristic polynomial, $f(x)$, used to construct the circuits is a primitive polynomial, and thus each LFSR can generate an *m*-sequence. Let

$$r(x) = f(x)^{-1} = x^nf(1/x). \quad (3)$$

Then, $r(x)$ is defined as a **reciprocal polynomial** of $f(x)$ [6]. A reciprocal polynomial of a primitive polynomial is also a primitive polynomial. Hence, if the reciprocal polynomial of $f(x)$ is used to construct a standard or modular LFSR with $r(x) = 1 + x^2 + x^3 + x^4 + x^5$, then the LFSR can also generate an *m*-sequence.

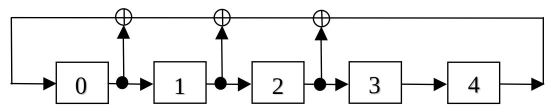


Figure 3. A 5-stage standard LFSR implementing $f(x) = 1+x^2+x^3+x^4+x^5$.

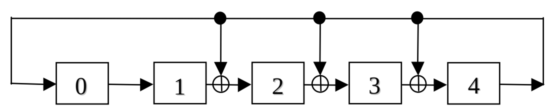


Figure 4. A 5-stage modular LFSR implementing $f(x) = 1+x^2+x^3+x^4+x^5$.

2.4 Hybrid LFSRs

Let a polynomial over GF(2), $1 + a(x) = b(x) + c(x)$, be said to be **fully decomposable** iff both $b(x)$ and $c(x)$ have no common terms and there exists an integer j such that $c(x) = x^jb(x)$, where $j \geq 1$. For example, if $1 + f(x)$ is fully decomposable such that

$$f(x) = 1 + b(x) + x^jb(x) \quad (4)$$

then a **(hybrid) top-bottom LFSR** [9] can be constructed using the feedback connection notation

$$s(x) = 1 + \hat{}x^j + x^jb(x) \quad (5)$$

where $\hat{}x^j$ indicates that the XOR gate with one input taken from the *j*th stage output of the LFSR is connected to the feedback path, not between stages. Similarly, if $f(x) + x^n$ is fully decomposable such that

$$f(x) = b(x) + x^jb(x) + x^n \quad (6)$$

then a **(hybrid) bottom-top LFSR** [9] can be constructed using the feedback connection notation

$$s(x) = b(x) + \hat{}x^{n-j} + x^n. \quad (7)$$

Assume a maximum-length LFSR uses k 2-input XOR gates to generate an *m*-sequence. It was shown in [9] that *if* $1 + f(x)$ *or* $f(x) + x^n$ *for constructing a standard or modular LFSR is fully decomposable, then a hybrid LFSR* *can be realized with only (k+1)/2 XOR gates. Also, if a top-bottom LFSR exists for f(x), then a bottom-top LFSR will exist for its reciprocal polynomial r(x), and vice versa.*

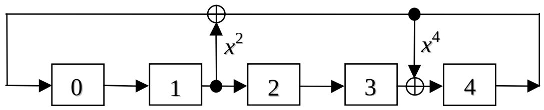


Figure 5. A 5-stage top-bottom LFSR using $s(x) = 1+{}^\wedge x^2+x^4+x^5$ to implement $f(x) = 1+x^2+x^3+x^4+x^5$.

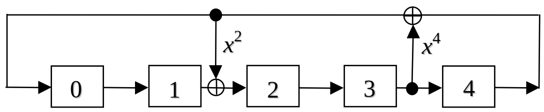


Figure 6. A 5-stage bottom-top LFSR using $s(x) = 1+x^2+{}^\wedge x^4+x^5$ to implement $f(x) = 1+x+x^2+x^3+x^5$.

Fig. 5 shows an example 5-stage top-bottom LFSR. The circuit implements the same $f(x)$, $1 + x^2 + x^3 + x^4 + x^5$, as that for Figs. 3 and 4. Since $f(x) = 1 + (x^2+x^3) + x^2(x^2+x^3)$, by Eq. 5, $s(x) = 1 + {}^\wedge x^2 + x^2(x^2+x^3) = 1 + {}^\wedge x^2 + x^4 + x^5$. As $f(x)$ is a primitive polynomial, the top-bottom LFSR will generate an *m*-sequence.

Fig. 6 shows a bottom-top LFSR that implements the reciprocal polynomial, $1+x+x^2+x^3+x^5$, of the primitive polynomial for Fig. 5. Since $f(x) = (1+x^2) + x(1+x^2) + x^5$, by Eq. 7, $s(x) = (1+x^2) + {}^\wedge x^{5\text{-}1} + x^5 = 1 + x^2 + {}^\wedge x^4 + x^5$. As a reciprocal polynomial of a primitive polynomial is a primitive polynomial, the bottom-top LFSR will also generate an *m*-sequence.

As can be seen, each circuit illustrated in Figs. 5 and 6 uses only two 2-input XOR gates, rather than three XOR gates for Figs. 3 and 4. Assume *k* XOR gates are required to implement a standard LFSR or a modular LFSR to produce an *m*-sequence, where the integer *k* must be an odd number. The hybrid LFSR design will require only $(k+1)/2$ 2-input XOR gates. Since the feedback path of the hybrid LFSR will drive fewer fanout nodes than that of the standard or modular LFSR, the hybrid design will have better operating performance.

3. Ring Generator Designs

One common drawback of using the standard LFSR, modular LFSR, and hybrid LFSR to generate pseudorandom bit sequences is the long delay associated with the feedback path. In the standard LFSR case, data at the output of the rightmost flip-flop would need to pass through *k* 2-input XOR gates to reach the leftmost flip-flop. In the modular LFSR case, the rightmost flip-flop would need to be strong enough to drive k+1 (fanout) nodes. In the hybrid LFSR case, the rightmost flip-flop would need to pass through one 2-input XOR gate before or after driving $(k+1)/2$ fanout nodes. Combined with their respective irregularity in design style, these types of LFSR designs may have difficulty to meet frequency requirement for high-performance applications.

3.1 Ring Generators and Hybrid Ring Generators

Consider the circuit given in Figs. 7-9. Each two adjacent flip-flops contain at most one 2-input XOR gate and each flip-flop output drives at most 2 fanout nodes. The circuit is constructed in a ring structure so there is no long feedback path connecting the rightmost flip-flop to the leftmost flip-flop. A circuit in so constructed is referred to as a **ring generator** [10] (see Fig. 7). Since the XOR gates are placed on the top and bottom rows simultaneously, a ring generator constructed with this additional property is referred to as a **hybrid ring generator**. Also, if the first XOR gate connecting to the leftmost stages is placed on the top row, then the hybrid ring generator is referred to as a **(hybrid) top-bottom ring generator** (see Fig. 8). Similarly, if the first XOR gate connecting to the leftmost stages is placed on the bottom row, then the hybrid ring generator is referred to as a **(hybrid) bottom-top ring generator** (see Fig. 9).

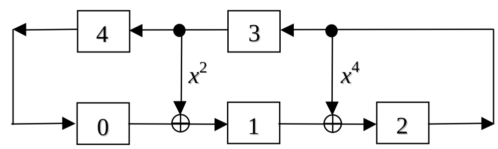


Figure 7. A 5-stage ring generator implementing $f(x) = 1+x^2+x^4+x^5$ (not a primitive polynomial).

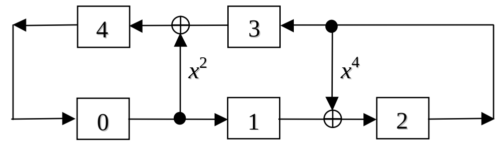


Figure 8. A 5-stage top-bottom ring generator constructed by $s(x) = 1+{}^\wedge x^2+x^4+x^5$ given in Fig. 5.

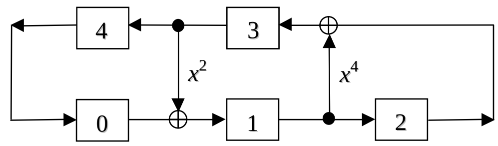


Figure 9. A 5-stage bottom-top ring generator constructed by $s(x) = 1+x^2+{}^\wedge x^4+x^5$ given in Fig. 6.

In more specific, a ring generator or a hybrid ring generator constructed either in a top-bottom or bottom-top form, exhibits the following properties:

1. Every output of a flip-flop in the design will drive at most 2 fanout nodes.
2. There will be at most one 2-input XOR gate placed between any two flip-flops, and thus each output signal of any flip-flop will only have to propagate through at most one 2-input XOR gate.
3. There will be no long feedback path, as the circuit is implemented in a ring structure.
4. Its regular and modular structure will result in simplified layout and routing, making the circuit timing and layout friendly.
5. The numbers of 2-input XOR gates used in the ring generator and the hybrid ring generator will be *k* and $(k+1)/2$, respectively.

3.2 Transformed LFSRs

Consider the circuit given in Fig. 10 first. This circuit was taken from FIG. 14 of [12] to illustrate a particular situation where it is required to add an *extra* 2-input XOR gate in a modular LFSR when *a source tap crossing a destination tap while moving to the left* (**SDL**) transformation is used to construct a transformed LFSR (t-LFSR) and where the inserted extra gate can cancel an available XOR gate, thereby reducing the number of XOR gates in the circuit by one.

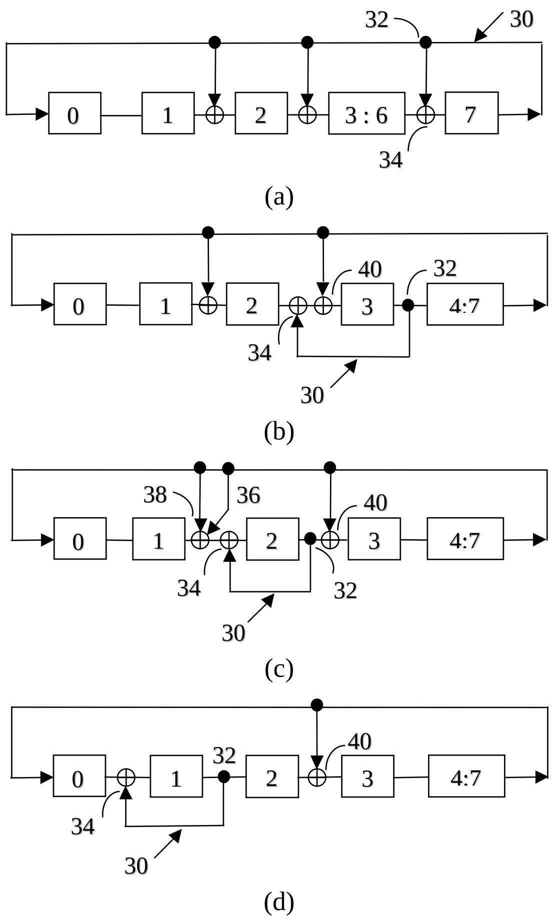


Figure 10. An 8-stage transformed LFSR constructed using the transformations given in [12] for $f(x) = 1+x^2+x^3+x^7+x^8$.

Fig. 10a shows a modular LFSR implementing $f(x) = 1 + x^2 + x^3 + x^7 + x^8$. First, an *elementary shift left* (**EL**) transformation is applied 4 times to the feedback connection represented by coefficient x^7 (feedback connection 30 with source tap 32 and destination gate 34). This leads to the circuit shown in Fig. 10b. Next, transformation SDL is applied to shift the feedback connection 30 further to the left by one flip-flop and adds a feedback connection line 36 at the input to the XOR gate 34 as shown in Fig. 10c. Because another XOR gate 38 with the same connectivity already exists at the output of flip-flop 1, the XOR gate 34 and connection 36 can be discarded. This reduces the number of XOR gates in the LFSR from 3 to 2. To reduce the load of flip-flop 2 that drives XOR gates 40 and 34 in Fig. 10c, an additional transformation EL is applied in Fig. 10d that shifts the feedback connection 30 further to the left. As a result, the transformed LFSR uses only 2 XOR gates and every flip-flop output drives at most two fanout nodes.

4. Minimum-Cost LFSRs

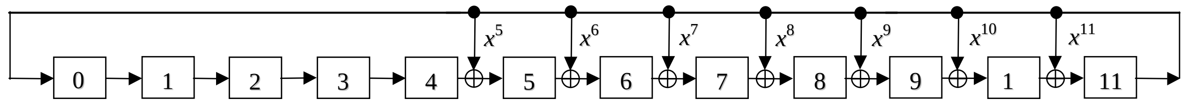
Up to this point, we mainly survey LFSR-based designs that implement primitive polynomials to illustrate the importance of generating *m*-sequences for specific applications. In reality, all these designs are applicable to implement non-primitive polynomials.

One issue that remains to be answered is what the true minimum hardware cost in each LFSR-based design is, when it comes to the design of a hybrid LFSR, a hybrid ring generator, or a transformed LFSR which uses fewer than k 2-input XOR gates than its corresponding standard LFSR, modular LFSR, or ring generator, regardless of whether $f(x)$ is a primitive polynomial or not. We will answer the question in this section by giving a new class of **minimum-cost LFSRs** (*min*-LFSRs) that uses only m 2-input XOR gates when $k \leq 2^m - 1$, or m+1 2-input XOR gates when $k \leq 2^m$, and then give proofs that $\log_2(k+1)$ when k is an odd number or $1+\log_2 k$ when k is an even number is the minimum number of 2-input XOR gates in constructing an LFSR-based design for $k \geq 1$.

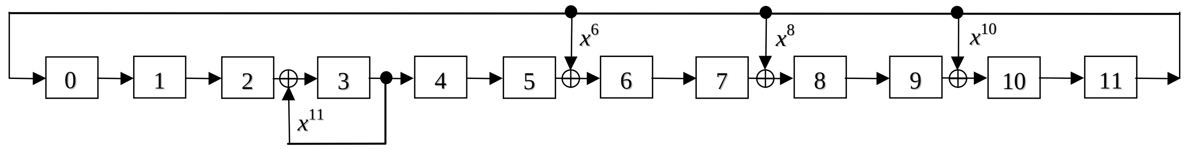
4.1 The Designs when $k = 2^m - 1$ or 2^m

Consider the 12-stage modular LFSR given in Fig. 11a. The circuit implements a non-primitive characteristic polynomial $f(x) = 1 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$, where $k = 7 = 2^m - 1 = 2^3 - 1$. $m = 3$. A primitive polynomial having a similar property is $f(x) = 1 + x^9 + x^{17} + x^{26} + x^{34} + x^{43} + x^{51} + x^{60} + x^{68}$ which is the reciprocal polynomial of a primitive polynomial of degree 68 listed in [13].

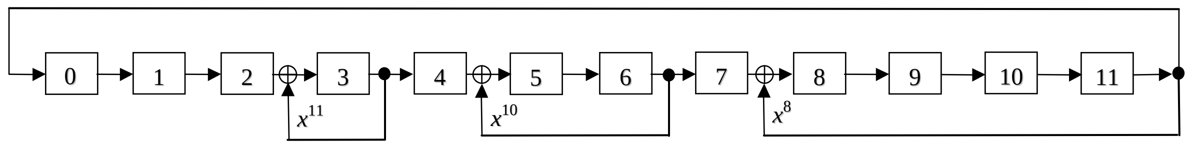
Fig. 11b shows a first transformed LFSR after applying transformations EL and SDL on the x^{11} arc to Fig. 11a. The combined $\{x^{11}, x^{10}\}$ arcs cancelled the x^9 arc; the $\{x^{11}, x^8\}$ arcs cancelled the x^7 arc; and the $\{x^{11}, x^6\}$ arcs cancelled the x^5 arc. Fig. 11c shows a second transformed LFSR after further applying transformations EL and SDL on the x^{10} arc to Fig. 11b. The combined $\{x^{10}, x^8\}$ arcs cancelled the x^6 arc. As a result, the final transformed LFSR shown in Fig. 11c contains only 3 arcs $\{x^{11}, x^{10}, x^8\}$ in the given order or uses only $m = 3$ 2-input XOR gates. This is in sharp contract to the modular LFSR given in Fig. 11a which uses $k = 7$ 2-input XOR gates. Also, all arcs in $\{x^{11}, x^{10}, x^8\}$ have a distance of {12-11, 12-10, 12-8} = {1, 2, 4} relative to the rightmost stage output (x^{12}), respectively, and form a *disjoint* structure where no arc is included in another arc and the destination taps of all arcs point to the left.



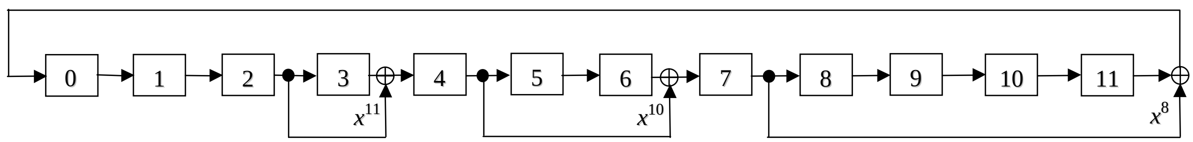
(a) A 12-stage modular LFSR with k = 7.



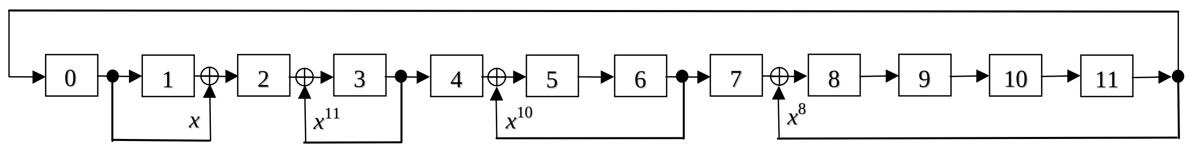
(b) A 12-stage transformed LFSR from (a) with k = 7.



(c) A 12-stage *min*-LFSR transformed from (b) with k = 7.



(d) A 12-stage dual *min*-LFSR of (c) with k = 7.



(e) A 12-stage *min*-LFSR with k = 8.

Figure 11. 12-stage transformed LFSRs toward *min*-LFSRs.

Looking into this non-primitive polynomial $f(x)$ further, one may find $1 + f(x)$ is *fully decomposable* such that $f(x) = 1 + x^5(1+x)(1+x^2)(1+x^4) = 1 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$. The coefficients i's of the 3 factored polynomials of $(1+x^i)$'s satisfy the following conditions: 1 < 2 and (1 + 2) < 4. If the coefficient of the x^5 term (which is 5) is greater than m, then the resultant circuit will be more modular because no flip-flop outputs will drive more than one XOR gate. The *min*-LFSR which is an equivalent circuit of Fig. 11c is shown in Fig. 12.

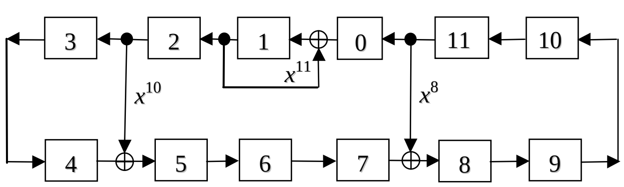


Figure 12. A 12-stage *min*-LFSR when k = 7.

Let $\boldsymbol{X} = \{x_0 \dots x_{11}\}$ and $\boldsymbol{Z} = \{z_0 \dots z_{11}\}$ represent the circuit's present state and next state, respectively. Linear equations over GF(2) governing the operation of Fig. 12 can be expressed as follows:

$$\begin{array}{ll} z_0 = x_{11} & z_1 = x_0 \\ z_2 = x_1 & z_3 = x_2 + x_3 \\ z_4 = x_3 & z_5 = x_4 + x_6 \\ z_6 = x_5 & z_7 = x_6 \\ z_8 = x_7 + x_{11} & z_9 = x_8 \\ z_{10} = x_9 & z_{11} = x_{10} \end{array} \qquad (8)$$

The set of linear equations can be further described by:

$$\boldsymbol{Z} = \boldsymbol{M} * \boldsymbol{X} \qquad (9)$$

or

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \\ z_{11} \end{bmatrix} = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \end{bmatrix} \qquad (10)$$

where matrix $\boldsymbol{M}$ is simply a **companion matrix** [6] whose characteristic polynomial $f(x)$ is defined as the **determinant** of $\boldsymbol{M} - \boldsymbol{I}x$, or symbolically:

$$f(x) = |\, \boldsymbol{M} - \boldsymbol{I}\boldsymbol{x} \,| \qquad (11)$$

Then, Eq. 11 can be rewritten as:

$$f(x) = \begin{vmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1+x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x \end{vmatrix} \qquad (12)$$

This yields $f(x) = 1 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$ which is the same $f(x)$ as one used to construct the modular LFSR shown in Fig. 11a. As can be seen, the *min*-LFSR uses only 3 2-input XOR gates, however, its design is not as modular as the hybrid designs shown in Figs. 7-9.

A similar *disjoint* circuit structure exists in the primitive polynomial, $f(x) = 1 + x^9 + x^{17} + x^{26} + x^{34} + x^{43} + x^{51} + x^{60} + x^{68}$ with $k = 2^3 - 1 = 7$. Applying transformations EL and SDL to the 68-stage modular LFSR that implements $f(x)$, the resultant transformed LFSR will contain 3 arcs $\{x^{60}, x^{51}, x^{34}\}$ each having a distance of {68-60, 68-51, 68-34} = {8, 17, 34} relative to the rightmost stage output (x^{68}), respectively. This means $1 + f(x)$ is *fully decomposable* such that $f(x) = 1 + x^9(1+x^8)(1+x^{17})(1+x^{34})$. The coefficients *i*'s of the 3 factored polynomials $(1+x^i)$'s also satisfy the following conditions: 8 < 17 and (8 + 17) < 34. Also, the coefficient of the x^9 term is greater than *m* (which is 3) to make the circuit more modular.

The above examples mainly illustrate how a *min*-LFSR is transformed from a corresponding modular LFSR. In fact, the same results can be achieved when a standard LFSR is used to implement the reciprocal polynomial $r(x)$ of $f(x)$ when the chosen $f(x)$ has resulted in a *min*-LFSR through transformations starting with a modular LFSR. In this case, the 12-stage standard LFSR with $k = 7$ shall implement $r(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^{12} = (1+x)(1+x^2)(1+x^4) + x^{12}$. A corresponding *min*-LFSR is shown in Fig. 11d with all transformed arcs now reversed and pointed to the right (not left). The circuit shown in Fig. 11d is referred to as a **dual LFSR** of that for Fig. 11c, and vice versa. The 68-stage standard LFSR shall now implement $r(x) = 1 + x^8 + x^{17} + x^{25} + x^{34} + x^{42} + x^{51} + x^{59} + x^{68} = (1+x^8)(1+x^{17})(1+x^{34}) + x^{68}$.

To further explore the transformation property of all hybrid designs discussed above, consider the circuits shown in Figs. 13a to 13d which are equivalent circuits of the hybrid designs shown in Figs. 5, 6, 8, and 9, respectively. One can see when $k = 3 = 2^2 - 1$, both $\{x^4, x^2\}$ arcs in each hybrid design also form a *disjoint* structure. These hybrid designs have been shown to have used a minimum of 2 2-input XOR gates according to Theorem 1 given in [11] when $k = 3$. Figs. 13a and 13c were obtained from their corresponding modular LFSRs, while Figs. 13b and 13d were obtained from their corresponding standard LFSRs. This leads to the following lemma:

Lemma 1: *Let* $k = 2^m - 1$. *Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, if 1 + f(x) or f(x) +* x^n *is fully decomposable such that*

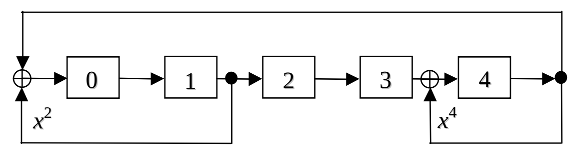
$$f(x) = 1 + x^a(1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) \qquad (13)$$

or

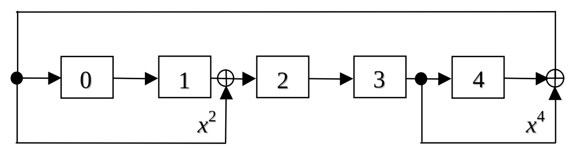
$$f(x) = (1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) + x^n \qquad (14)$$

and ***there are exactly m polynomials of*** $(1+x^{bi})$, *then a minimum-cost LFSR (min-LFSR) that implements the same f(x) as the standard or modular LFSR can be constructed using m 2-input XOR gates, where* $a \geq 1$, $b_1 < b_2$, $(b_1 + b_2) < b_3$, …, $(b_1 + b_2 + \ldots + b_{m-1}) < b_m$, $(b_1 + b_2 + \ldots + b_{m-1} + b_m) < n$.

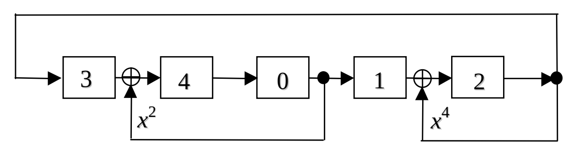
Proof: See previous discussion in this section. In addition, if $a \geq m$ in Eq. 13 or $(b_1 + b_2 + \ldots + b_m) \leq (n - m)$ in Eq. 14 holds, then the structure of the *min*-LFSR will be more modular. Because $k = 2^m - 1$, the *min*-LFSR will use $\log_2(k+1)$ 2-input XOR gates. □



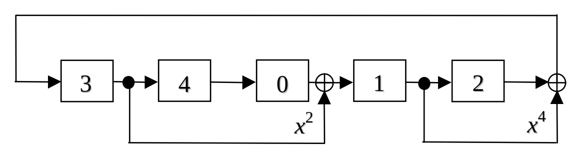
(a) Equivalent top-bottom LFSR of Fig. 5



(b) Equivalent bottom-top LFSR of Fig. 6



(c) Equivalent top-bottom ring generator of Fig. 8



(d) Equivalent bottom-top ring generator of Fig. 9

Figure 13. Equivalent circuits of hybrid designs.

Now consider the case when $k = 2^m$. Let $f(x) = 1 + x + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$ with $k = 2^m = 2^3 = 8$. Because $f(x)$ can be factored such that $f(x) = (1+x) + x^5(1+x)(1+x^2)(1+x^4)$, the resultant transformed LFSR will contain (m+1) = 4 arcs {^x, x^{11}, x^{10}, x^8}. The 3 arcs in {x^{11}, x^{10}, x^8} have a distance of {12-11, 12-10, 12-8} = {1, 2, 4} relative to the rightmost stage output (x^{12}), respectively. The ^x arc have a distance of 1 relative to the leftmost stage input (x^0). The 4 arcs also form a *disjoint* structure with the destination tap of the ^x arc pointing to the right, and the destination taps of the other three arcs {x^{11}, x^{10}, x^8} pointing to the left. The transformed LFSR is shown in Fig. 11e. Its equivalent circuit is shown in Fig. 14.

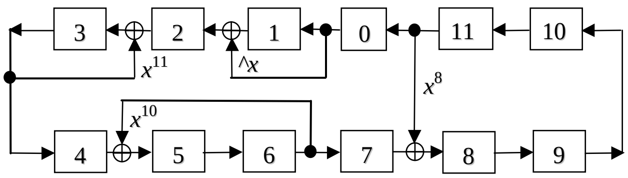


Figure 14. A 12-stage *min*-LFSR when k = 8.

Similarly, a *min*-LFSR LFSR with $k = 2^m$ can be also used to implement the reciprocal polynomial $r(x)$ of the $f(x)$ which has resulted in a *min*-LFSR through transformations starting with a modular LFSR. In this case, the 12-stage standard LFSR with k = 8 shall implement $r(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^{11} + x^{12} = (1+x)(1+x^2)(1+x^4) + (x^{11}+x^{12})$. Its corresponding *min*-LFSR (not shown) will be similar to Fig. 11e but with all transformed arcs now reversed and pointed to the right (not left). This leads to the following lemma:

Lemma 2: *Let $k = 2^m$. Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, if f(x) can be factored such that*

$$f(x) = (1+x^c) + x^a(1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) \qquad (15)$$

or

$$f(x) = (1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) + (x^{n-c}+x^n) \qquad (16)$$

and ***there are exactly m polynomials of ($1+x^{bi}$)****, then a minimum-cost LFSR (min-LFSR) that implements the same f(x) as the standard or modular LFSR can be constructed using m+1 2-input XOR gates, where $c < a$, $b_1 < b_2$, $(b_1 + b_2) < b_3$, …, $(b_1 + b_2 + \ldots + b_{m-1}) < b_m$, $(b_1 + b_2 + \ldots + b_m) < n\text{-}c$.*

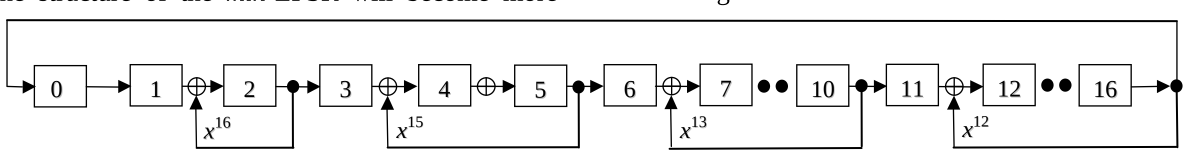
Proof: See previous discussion in this section. In addition, if (a-c) > m in Eq. 15 or $c < m$ in Eq. 16 holds, then the structure of the *min*-LFSR will become more modular. Because $k = 2^m$, the *min*-LFSR will 1+$\log_2 k$ 2-input XOR gates. □

Note that a *min*-LFSR in so constructed cannot generate an *m*-sequence, because $f(x)$ is a not primitive polynomial. A primitive polynomial has an inherent property that k must be always an odd number. That is, while both lemmas are provided for construction of a *min*-LFSR that will yield the lowest hardware cost, the characteristic polynomial chosen to construct the *min*-LFSR does not necessarily implement a primitive polynomial.

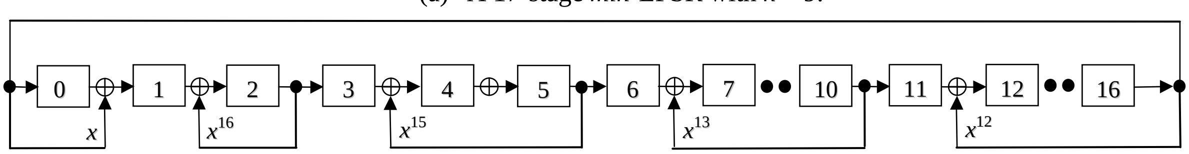
4.2 The Designs when $k \neq 2^m$ - 1 or 2^m

In case $k \neq 2^m - 1$ or 2^m, a transformed LFSR can still be in a *disjoint* structure. For example, let $f(x) = 1 + x^5(1+x)(1+x^2)(1+x^4)(1+x^5) = 1 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{13} + x^{14} + x^{15} + x^{16} + x^{17}$ with k = 9. The resultant 17-stage *min*-LFSR is shown in Fig. 15a. The circuit contains 4 arcs {x^{16}, x^{15}, x^{13}, x^{12}} each having a distance of {1, 2, 4, 5} relative to the output of flip-flop 16, thereby causing the *min*-LFSR to use 4 XOR gates. Transformations on the t-LFSR are complex that involve creation of three news arcs {x^{12}, x^{11}, x^{10}} by the x^{16} feedback tap, and subsequent cancellation of the {x^{11}, x^{10}} arcs by the x^{15} feedback tap. One major restriction on $f(x)$ with $k < 2^m - 1$ is that the highest coefficient of the x^5 term in $(1+x^5)$ cannot be greater than the sum of the coefficients of all other x^i terms in $(1+x^i)$'s, *i.e.*, 5 < (1+2+4). This will allow creation and cancellation of new arcs. Fig. 15b further illustrates how the 5 arcs in {^x, x^{16}, x^{15}, x^{13}, x^{12}} form a *disjoint* structure for a *min*-LFSR that implements $f(x) = 1 + x + x^5(1+x)(1+x^2)(1+x^4)(1+x^5) = 1 + x + x^5 + x^6 + x^7 + x^8 + x^9 + x^{13} + x^{14} + x^{15} + x^{16} + x^{17}$ with k = 10.

Similarly, a *min*-LFSR LFSR can be also used to implement the reciprocal polynomial $r(x)$ of the $f(x)$ which has resulted in a *min*-LFSR through transformations starting with a modular LFSR. In this case, the 17-stage standard LFSR with k = 9 shall implement $r(x) = (1+x)(1+x^2)(1+x^4)(1+x^5) + x^{17} = 1 + x + x^2 + x^3 + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{17}$, whereas the 17-stage standard LFSR with k = 10 shall implement $r(x) = (1+x)(1+x^2)(1+x^4)(1+x^5) + (x^{16}+x^{17}) = 1 + x + x^2 + x^3 + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{16} + x^{17}$. This leads to the following two lemmas:



(a) A 17-stage *min*-LFSR with k = 9.



(b) A 17-stage *min*-LFSR with k = 10.

Figure 15. 17-stage transformed LFSRs toward *min*-LFSRs.

Lemma 3: *Let p* = 2^m **–** 1. *Let p be the smallest integer greater than or equal to k, where k is an odd number. Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, if 1 + f(x) or f(x) +* x^n *is fully decomposable such that*

$$f(x) = 1 + x^a(1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) \quad (17)$$

or

$$f(x) = (1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) + x^n \quad (18)$$

and ***there are exactly m polynomials of*** $(1+x^{bi})$, *then a minimum-cost LFSR (min-LFSR) that implements the same f(x) as the standard or modular LFSR can be constructed using m 2-input XOR gates, where* $a \geq 1$, $b_1 < b_2$, $(b_1 + b_2) < b_3$, …, $(b_1 + b_2 + \ldots + b_{m-2}) < b_{m-1}$, $\mathbf{(b_1 + b_2 + \ldots + b_{m-1}) \geq b_m}$, $(b_1 + b_2 + \ldots + b_{m-1} + b_m) < n$.

Proof: See previous discussion in this section. In addition, if $a \geq m$ in Eq. 17 or $(b_1 + b_2 + \ldots + b_m) \leq (n - m)$ in Eq. 18 holds, then the structure of the *min*-LFSR will be more modular. □

Lemma 4: *Let p* = 2^m. *Let p be the smallest integer greater than or equal to k, where k is an even number. Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, if f(x) can be factored such that*

$$f(x) = (1+x^c) + x^a(1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) \quad (19)$$

or

$$f(x) = (1+x^{b1})(1+x^{b2})\ldots(1+x^{bm}) + (x^{n-c}+x^n) \quad (20)$$

and ***there are exactly m polynomials of*** $(1+x^{bi})$, *then a minimum-cost LFSR (min-LFSR) that implements the same f(x) as the standard or modular LFSR can be constructed using m+1 2-input XOR gates, where* $c < a$, $b_1 < b_2$, $(b_1 + b_2) < b_3$, …, $(b_1 + b_2 + \ldots + b_{m-2}) < b_{m-1}$, $\mathbf{(b_1 + b_2 + \ldots + b_{m-1}) \geq b_m}$, $(b_1 + b_2 + \ldots + b_m) < n\text{-}c$.

Proof: See previous discussion in this section. In addition, if $(a\text{-}c) > m$ in Eq. 19 or $c < m$ in Eq. 20 holds, then the structure of the *min*-LFSR will become more modular. □

Lemmas 3 and 4 imply that there exists a *min*-LFSR that uses only $\log_2(k+1)$ 2-input XOR gates when *k* is an odd number, or $1+\log_2 k$ when *k* is an even number. As an example, Table 1 lists the number of 2-input XOR gates used for *k* 1 through 16 in each LFSR-based design. The table shows that *if an odd number k results in an m value in a min-LFSR, then an even number k+1 will produce an m+1 value*.

We now give proofs that any LFSR-based design cannot use fewer than $\log_2(k+1)$ 2-input XOR gates when *k* is an odd number or $1+\log_2 k$ 2-input XOR gates when *k* is an even number.

Theorem 1: *Let k be an odd number. Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, a transformed LFSR (t-LFSR) that implements the same f(x) as the standard or modular LFSR cannot use fewer than* $\log_2(k+1)$ *2-input XOR gates.*

Proof: We prove the theorem by satisfying the necessary and sufficient conditions. By Lemmas 1 and 3, we have shown that a t-LFSR using only $\log_2(k+1)$ 2-input XOR gates can be constructed to implement the same $f(x)$ as a standard or modular LFSR that uses *k* 2-input XOR gates when *k* is an odd number. Hence, the necessary condition is satisfied.

Table 1. Number of 2-Input XOR Gates for each (*k*, *m*)

Standard LFSR Modular LFSR (*k*)	Ring Generator (*k*)	Hybrid LFSR Hybrid Ring Generator (*k* + 1) / 2	*min*-LFSR (*m* or *m*+1)
1	**1**	**1**	**1**
2	-	-	2
3	**3**	**2**	**2**
4	-	-	3
5	5	3	3
6	-	-	4
7	**7**	**4**	**3**
8	-	-	4
9	9	5	4
10	-	-	5
11	11	6	4
12	-	-	5
13	13	7	4
14	-	-	5
15	**15**	**8**	**4**
16	-	-	5

We now prove the sufficient condition by contradiction. Assume the t-LFSR forms a *disjoint* structure that contains *m* distinct transformed arcs. If any of the transformed arc were cancelled by any combination of two other arcs, the resultant t-LFSR would contain only *m*-1 disjoint transformed arcs. By retransforming these *m*-1 disjoint arcs in the t-LFSR back to a standard or modular LFSR, the standard or modular LFSR would use less than *k* (no more than 2^{m-1}-1) 2-input XOR gates. This means the circuit would have implemented a different $f_2(x)$. This contradicts the condition that the t-LFSR must implement the same $f(x)$ as the given standard or modular LFSR. This concludes the proof. □

Theorem 2: *Let k be an even number. Given f(x) that constructs an n-stage standard or modular LFSR with k 2-input XOR gates, a transformed LFSR (t-LFSR) that implements the same f(x) as the standard or modular LFSR cannot use fewer than* $1+\log_2 k$ *2-input XOR gates.*

Proof: Similar to Theorem 1, Lemmas 2 and 4 can be used instead to conclude the proof. □

5. Construction Method

To better understand how a *min*-LFSR can be designed via visual inspection or by a construction method, consider the 12-stage *min*-LFSR illustrated in Fig. 14 for implementing $f(x) = 1 + x + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} = (1+x)(1+x^2)(1+x^4) + (x^{11}+x^{12})$ with $k = 8$ again.

The *min*-LFSR contains 4 arcs $\{\wedge x, x^{11}, x^{10}, x^8\}$. The 3 x^j arcs in $\{\wedge x, x^{11}, x^{10}, x^8\}$ are first renumbered to $\{\wedge x, x^1, x^2, x^4\}$ based on their relative distance to flip-flop 11. Fig. 16 is an isomorphic circuit of Fig. 14 by further renumbering the flip-flops from 0 to 11 *counterclockwise* beginning with the leftmost bottom flip-flop.

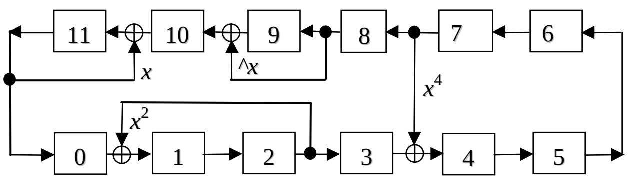


Figure 16. A 12-stage isomorphic *min*-LFSR with $k = 8$.

Assume the *min*-LFSR can be made more modular when its $f(x)$ satisfies one of the conditions given in Eqs. 13-20. A **visual inspection** method to design such a *min*-LFSR is now given below:

Step 1: Select a (primitive) polynomial of degree n as the characteristic polynomial $f(x)$ such that it can be result in Eq. 13 or 17 when k is an odd number or Eq. 15 or 19 when k is an even number; Let the transformed taps in each $(1+x^j)$ be $\{\wedge x^c, x^{b1}, x^{b2}, \ldots, x^{bm}\}$.

Step 2: Place half (or one less) of the flip-flops on the top row and the rest of the flip-flops on the bottom row and then stitch them together to form a ring structure;

Step 3: Label the flip-flop numbers from 0 to n–1 *counterclockwise*, always beginning with the leftmost bottom flip-flop;

Step 4: Create a feedback connection for tap x^{bm} on the bottom row by encompassing b_m adjacent flip-flops, beginning with the rightmost ones;

Step 5: For tap $\wedge x^c$ when $k = 2^m$, create a feedback connection that has a distance of c and place one 2-input XOR gate with the $\wedge x^c$ arc pointed against the x^{bm} tap.

Step 6: for each of the remaining x^j taps, create *in succession* a feedback connection that has a distance of j and place one additional 2-input XOR gate, where $j < b_m$, starting with tap x^{b1} first.

Step 7: Reverse the directions of all taps to create a dual *min*-LFSR if the circuit implements a reciprocal polynomial of $f(x)$ or Eq. 14, 16, 18, or 20.

The positions of the source and destination taps of each arc in the *min*-LFSR can also be calculated using the following **construction method**:

Step 1: Let T_i represent the distance of the ith tap to the rightmost stage in a modular LFSR by $\{x^{bm}, \wedge x^c, x^{b1}, x^{b2}, \ldots, x^{bm-1}\}$, $i \geq 1$; S_i and D_i indicate the locations of the source and destination taps (as inputs to a 2-input XOR gate) in the resultant *min*-LFSR, respectively; and L be the number of flip-flops in a *min*-LFSR; calculate locations of the source and destination taps according to the following formulas:

$$S_i = (S_{i-1} + T_i + 1) \bmod L \qquad (21)$$
$$D_i = (S_{i-1} + 1) \bmod L. \qquad (22)$$

with an initial condition: $S_0 = (L - T_1) / 2 - 2$.

Consider Fig. 16 again. $L = 12$. The circuit contains 4 arcs $\{\wedge x, x, x^2, x^4\}$. These 4 arcs are first reordered to $\{x^4, \wedge x, x, x^2\}$ according to Step 1. The reason is because in so doing, we will draw a vertical line (with a much shorter wire length) for the x^4 arc that has the longest distance. Also, we may be able to draw another vertical line for the x^2 arc that has the second longest distance to further reduce the overall wire length (as shown in Fig. 12). These 4 arcs are now represented by a sequence $T_1 = 4$, $T_2 = 1$, $T_3 = 1$, $T_4 = 2$. Thus, using the above formulas will yield the following feedback connections: $S_0 = (12–4)/2 – 2 = 2$; $S_1 = (2+4+1) \bmod 12 = \mathbf{7}$, $D_1 = (2+1) \bmod 12 = \mathbf{3}$; $S_2 = (7+1+1) \bmod 12 = \mathbf{9}$, $D_2 = (7+1) \bmod 12 = \mathbf{8}$**;** $S_3 = (9+1+1) \bmod 12 = \mathbf{11}$, $D_3 = (9+1) \bmod 12 = \mathbf{10}$**;** $S_4 = (11+2+1) \bmod 12 = \mathbf{2}$, $D_4 = (11+1) \bmod 12 = \mathbf{0}$. The 4 taps can be expressed as a list of pairs: (7,3), (**9,8**), (11,10), (2,0).

Step 2: Reverse the direction of the tap to create the $\wedge x^c$ tap.

For example, since (S_2, D_2) represents the original $\wedge x$ taps, the above pair list now becomes (7,3), (**8,9**), (11,10), (2,0). You may now verify the feedback connections in Fig. 16.

Three sets of primitive polynomials each consisting of 5, 9, or 17 terms [*a.k.a.* weights, exponents, or coefficients] of degree up to 800 that meet the *fully decomposable* requirement given in Eq. 14 are listed in Appendices 1 to 3, respectively. These primitive polynomials were found using modified NTL and Magma programs [14, 15]. Minimum-weight primitive polynomials with $k = 1$ or 3 can also be found in the Appendix [11].

We formulated the search according to the following formulas:

For $k = 3$:

$$p(x) = (1 + x^a)(1 + x^b) + x^n \qquad (23)$$

where $1 \leq a < b < n$, $(a + b) < n$.

For $k = 7$:

$$p(x) = (1 + x^a)(1 + x^b)(1 + x^c) + x^n \qquad (24)$$

where $1 \leq a < b < c < n$, $(a + b) < c$, $(a + b + c) < n$.

For $k = 15$:

$$p(x) = (1 + x^a)(1 + x^b)(1 + x^c)(1 + x^d) + x^n \qquad (25)$$

where $1 \leq a < b < c < d < n$; $(a + b) < c$, $(a + b + c) < d$, $(a + b + c + d) < n$.

We sped up the search by putting a constraint, $a \leq n/2$, on variable a, because if a $p(x)$ with $a \leq n/2$ does not exist, then its reciprocal polynomial with $a > n/2$ will not exist.

It is interesting to note that such primitive polynomials exist for every degree 5 through 800 when $k = 3$, every degree 12 through 800 when $k = 7$, and every degree 19 through 800 when $k = 15$. Based on the construction method, each polynomial listed in the Appendices can now be used to construct a *min*-LFSR.

6. Comparative Analysis

Table 2 summarizes the design features of various LFSR-based designs. The top-bottom (or bottom-top) LFSR will have one level (or two levels) of XOR logic because it is constructed to have *only* one 2-input XOR gate connected to the feedback path according to Eq. 5 (or Eq. 7). On the other hand, the feedback path in each top-bottom or bottom-top LFSR will always drive $(k+1)/2$ fanout nodes due to the nature of the design. As to *cellular automaton* (CA), in general, the total number of 2-input XOR gates used in a CA design will be equal to $2n$-2 for providing better randomness [16].

Table 2. Features of LFSR-Based Designs

	XOR Gates	Levels of Logic	Fanout
Standard LFSR	k	$\log_2 k$	2
Modular LFSR	k	1	$k + 1$
Top-Bottom LFSR	$(k + 1) / 2$	1	$(k + 1) / 2$
Bottom-Top LFSR	$(k + 1) / 2$	2	$(k + 1) / 2$
Cellular Automaton	$2n – 2$	2	3
Ring Generator	k	1	2
Hybrid Ring Generator	$(k + 1) / 2$	1	2
Minimum-Cost LFSR	$\log_2(k + 1)$, odd k	1	2
Minimum-Cost LFSR	$1 + \log_2 k$, even k	1	2

The authors showed in Theorem 1 [11] that *given a maximum-length standard or modular LFSR using k 2-input XOR gates, a modified LFSR implementing the same f(x) as the standard or modular LFSR can never use fewer than (k+1)/2 XOR gates, when k = 1, 3, or 5*. We found the results are the same as Theorem 1 given here. However, the combined Theorems 1 and 2 have provided much broad proofs for $k \geq 1$.

7. Conclusion

This paper showed by examples and gave proofs that given a standard or modular LFSR using k 2-input XOR gates, a minimum-cost LFSR (*min*-LFSR) can be designed to use a minimum number of $\log_2(k+1)$ 2-input XOR gates when k is an odd number or $1+\log_2 k$ 2-input XOR gates when k is an even number. These *min*-LFSRs exist only when $f(x)$ meets the *fully decomposable* requirement. The *min*-LFSR that implements the chosen characteristic polynomial, $f(x)$, however, can be a non-primitive polynomial. If a primitive polynomial of degree n with a particular k does not exist to construct an n-stage *min*-LFSR, one may consider using a *min*-LFSR with $k = 2^m$-1 that use the same number of XOR gates as the unavailable n-stage *min*-LFSR, because most likely primitive polynomials with $k = 2^m$-1 will exist for every degree up to 800, such as k = 3, 7, and 15.

8. Acknowledgments

The authors sincerely express our gratitude to Professor Samuel S. Wagstaff, Jr. in the Departments of Computer Sciences and Mathematics at Purdue University for providing the needed prime factors so we can use NTL for computations to generate desired primitive polynomials and check the results with those generated by Magma, or vice versa. We also would like to thank Alice Yu of the University of California at San Diego and Teresa Chang of SynTest Technologies for drawing all figures. This research was supported in part by the National Science Foundation under Grant No. CCF-0916837.

References

[1] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, MIT Press, Cambridge, Massachusetts, 1972.

[2] M.L. Bushnell and V.D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*, Springer, New York, 2000.

[3] L.-T. Wang, C.-W. Wu, and X. Wen, editors, *VLSI Test Principles and Architectures: Design for Testability*, Morgan Kaufmann, San Francisco, 2006.

[4] N.A. Touba, “Survey of Test Vector Compression Techniques,” *IEEE Design & Test of Computers*, pp. 294-303, July-Aug. 2006.

[5] W. Trappe and L.C. Washington, *Introduction to Cryptography with Coding Theory*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey, 2005.

[6] S.W. Golomb, *Shift Register Sequence*, Aegean Park Press, Laguna Hills, California, 1982.

[7] C. Arvillias and D.G. Maritsas, “Toggle-Registers Generating in Parallel k kth Decimations of m-sequences $X^p + X^k + 1$ Design Tables,” *IEEE Trans. on Computers*, vol. C-28, no. 2, pp. 89-101, Feb. 1979.

[8] W.W. Warlick and J.E. Hershey, “High-Speed *m*-Sequence Generators,” *IEEE Trans. on Computers*, vol. C-29, no. 5, pp. 398-400, May 1980.

[9] L.-T. Wang and E.J. McCluskey, “Hybrid Designs Generating Maximum-Length Sequences,” *IEEE Trans. on Computer-Aided Design*, vol. 7, no. 1, pp. 91-99, Jan. 1988.

[10] N. Mukherjee, J. Rajski, G. Mrugalski, A. Pogiel, and J. Tyszer, “Ring Generator: An Ultimate Linear Feedback Shift Register,” *IEEE Computer*, pp. 64-71, June 2011.

[11] L.-T. Wang, N.A. Touba, R.P. Brent, H. Wang, and H. Xu, “High-Speed Hybrid Ring Generator Design Providing Maximum-Length Sequences with Low Hardware Cost,” CERC Technical Report No. UT-CERC-12-01, Computer Engineering Research Center, University of Texas at Austin, Oct. 2011.

[12] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, “Method for Synthesizing Linear Finite State Machines,” United States Patent No. 6,353,842, March 5, 2002.

[13] N. Mukherjee, A. Pogiel, J. Rajski, and J. Tyszer, “High-Speed On-Chip Event Counters for Embedded Systems,” *Proc. IEEE Int. Conf. on VLSI Design*, pp. 275-280, 2009.

[14] NTL: http://www.shoup.net/ntl/.

[15] Magma: http://www.math.ufl.edu/help/magma/MAGMA.html.

[16] G. Mrugalski, J. Rajski, and J. Tyszer, “Cellular Automata-Based Test Pattern Generators with Phase Shifters,” *IEEE Trans. on Computer-Aided Design*, vol. 19, no. 8, pp. 878-893, Aug. 2000.

Appendix 1: 5-Weight Primitive Polynomials of Degree up to 800 over GF(2)

				5 2 1 0
6 3 1 0	7 2 1 0	8 5 1 0	9 3 1 0	10 3 1 0
11 5 1 0	12 4 3 0	13 3 1 0	14 11 1 0	15 6 1 0
16 3 2 0	17 2 1 0	18 7 3 0	19 5 1 0	20 11 1 0
21 15 1 0	22 6 1 0	23 4 1 0	24 3 1 0	25 2 1 0
26 7 1 0	27 7 1 0	28 20 1 0	29 6 1 0	30 15 1 0
31 2 1 0	32 27 1 0	33 7 1 0	34 14 1 0	35 7 1 0
36 7 1 0	37 10 2 0	38 5 1 0	39 21 1 0	40 19 2 0
41 2 1 0	42 22 1 0	43 5 1 0	44 26 1 0	45 3 1 0
46 20 1 0	47 4 1 0	48 27 1 0	49 8 1 0	50 26 1 0
51 15 1 0	52 24 1 0	53 15 1 0	54 36 1 0	55 20 1 0
56 21 1 0	57 36 1 0	58 5 1 0	59 21 1 0	60 13 3 0
61 15 1 0	62 56 1 0	63 4 1 0	64 3 1 0	65 3 1 0
66 9 1 0	67 9 1 0	68 25 1 0	69 27 2 0	70 15 1 0
71 8 1 0	72 47 6 0	73 17 1 0	74 15 1 0	75 10 1 0
76 35 1 0	77 30 1 0	78 19 1 0	79 8 1 0	80 37 1 0
81 27 1 0	82 35 3 0	83 45 1 0	84 78 1 0	85 27 1 0
86 12 1 0	87 52 1 0	88 71 1 0	89 26 1 0	90 18 1 0
91 83 1 0	92 12 1 0	93 31 1 0	94 5 1 0	95 16 1 0
96 47 2 0	97 32 1 0	98 7 1 0	99 45 2 0	100 81 1 0
101 6 1 0	102 76 1 0	103 8 1 0	104 10 1 0	105 6 1 0
106 5 1 0	107 63 2 0	108 42 1 0	109 6 1 0	110 12 1 0
111 39 1 0	112 43 2 0	113 8 1 0	114 81 1 0	115 14 1 0
116 70 1 0	117 18 2 0	118 59 1 0	119 30 1 0	120 111 7 0
121 47 1 0	122 59 1 0	123 13 1 0	124 78 1 0	125 107 1 0
126 36 1 0	127 47 1 0	128 27 2 0	129 4 1 0	130 71 1 0
131 47 1 0	132 27 1 0	133 51 1 0	134 26 1 0	135 70 1 0
136 125 1 0	137 14 1 0	138 7 1 0	139 5 3 0	140 44 1 0
141 31 1 0	142 84 1 0	143 20 1 0	144 69 1 0	145 5 1 0
146 59 1 0	147 37 1 0	148 17 1 0	149 109 1 0	150 118 1 0
151 2 1 0	152 65 1 0	153 24 1 0	154 127 2 0	155 31 1 0
156 115 1 0	157 26 1 0	158 26 1 0	159 148 1 0	160 18 1 0
161 15 1 0	162 87 1 0	163 59 1 0	164 13 1 0	165 30 1 0
166 38 1 0	167 34 1 0	168 15 2 0	169 21 1 0	170 151 1 0
171 18 1 0	172 132 1 0	173 99 1 0	174 135 1 0	175 132 1 0
176 118 1 0	177 84 1 0	178 75 1 0	179 33 1 0	180 36 1 0
181 6 1 0	182 127 1 0	183 52 1 0	184 101 1 0	185 30 1 0
186 22 1 0	187 57 1 0	188 73 1 0	189 126 1 0	190 17 1 0
191 8 1 0	192 27 1 0	193 14 1 0	194 71 1 0	195 9 1 0
196 65 1 0	197 61 1 0	198 45 1 0	199 45 1 0	200 41 1 0
201 16 1 0	202 167 1 0	203 7 1 0	204 73 1 0	205 29 1 0
206 28 1 0	207 72 1 0	208 59 3 0	209 66 1 0	210 32 3 0
211 45 1 0	212 22 1 0	213 7 1 0	214 48 1 0	215 75 1 0
216 195 1 0	217 11 1 0	218 7 1 0	219 18 1 0	220 14 1 0
221 34 1 0	222 91 1 0	223 32 1 0	224 30 1 0	225 24 1 0
226 57 1 0	227 45 1 0	228 147 1 0	229 63 1 0	230 45 1 0
231 18 1 0	232 99 1 0	233 25 1 0	234 49 1 0	235 9 1 0
236 68 1 0	237 25 1 0	238 167 1 0	239 13 1 0	240 119 2 0
241 84 1 0	242 131 1 0	243 75 1 0	244 39 1 0	245 167 1 0
246 34 1 0	247 21 1 0	248 63 2 0	249 246 1 0	250 99 1 0
251 227 1 0	252 160 1 0	253 6 1 0	254 18 1 0	255 93 1 0
256 99 1 0	257 44 1 0	258 121 1 0	259 14 1 0	260 20 1 0
261 63 1 0	262 96 1 0	263 12 1 0	264 9 1 0	265 14 1 0
266 6 1 0	267 85 1 0	268 15 1 0	269 6 1 0	270 171 2 0
271 53 1 0	272 107 1 0	273 6 1 0	274 69 1 0	275 22 1 0
276 88 1 0	277 69 1 0	278 4 1 0	279 4 1 0	280 41 1 0
281 94 1 0	282 157 1 0	283 59 1 0	284 36 1 0	285 105 1 0
286 129 1 0	287 76 1 0	288 10 1 0	289 71 1 0	290 133 1 0
291 105 2 0	292 114 1 0	293 95 1 0	294 43 1 0	295 12 1 0
296 33 1 0	297 4 1 0	298 29 1 0	299 46 1 0	300 101 3 0
301 65 1 0	302 50 1 0	303 28 1 0	304 195 1 0	305 12 1 0
306 225 1 0	307 115 2 0	308 296 1 0	309 154 1 0	310 15 1 0
311 30 1 0	312 305 3 0	313 113 1 0	314 175 1 0	315 9 1 0
316 99 1 0	317 95 1 0	318 114 1 0	319 128 1 0	320 3 1 0
321 13 1 0	322 21 2 0	323 203 1 0	324 255 1 0	325 75 1 0
326 89 1 0	327 94 1 0	328 91 2 0	329 74 1 0	330 15 1 0
331 323 1 0	332 12 1 0	333 53 3 0	334 26 1 0	335 41 1 0
336 211 1 0	337 132 1 0	338 103 1 0	339 193 1 0	340 92 1 0
341 23 1 0	342 84 1 0	343 20 1 0	344 259 1 0	345 15 1 0
346 179 1 0	347 337 1 0	348 127 1 0	349 11 1 0	350 120 1 0
351 132 1 0	352 75 1 0	353 58 1 0	354 118 1 0	355 5 1 0
356 48 1 0	357 69 1 0	358 332 1 0	359 169 1 0	360 25 1 0
361 44 1 0	362 26 1 0	363 7 1 0	364 44 1 0	365 71 1 0
366 24 1 0	367 11 1 0	368 113 1 0	369 79 1 0	370 315 1 0
371 15 1 0	372 195 1 0	373 99 1 0	374 63 1 0	375 7 1 0
376 141 1 0	377 17 1 0	378 301 1 0	379 113 1 0	380 17 1 0
381 183 2 0	382 174 1 0	383 22 1 0	384 163 1 0	385 65 1 0
386 85 1 0	387 67 1 0	388 68 1 0	389 153 1 0	390 151 1 0
391 21 1 0	392 345 1 0	393 96 1 0	394 153 2 0	395 269 1 0
396 231 1 0	397 66 1 0	398 100 1 0	399 49 1 0	400 117 1 0

Note: “12 4 3 0” means $p(x) = (1 + x^3)(1 + x^4) + x^{12} = 1 + x^3 + x^4 + x^7 + x^{12}$.

Appendix 1: 5-Weight Primitive Polynomials of Degree up to 800 over GF(2) – Cont'd

==

401 123 1 0	402 339 2 0	403 149 1 0	404 121 1 0	405 337 3 0
406 209 1 0	407 117 1 0	408 381 1 0	409 249 1 0	410 155 1 0
411 131 5 0	412 219 1 0	413 281 1 0	414 45 1 0	415 80 1 0
416 143 1 0	417 30 1 0	418 17 1 0	419 163 3 0	420 130 1 0
421 297 5 0	422 82 1 0	423 54 1 0	424 65 1 0	425 198 1 0
426 57 2 0	427 105 1 0	428 50 1 0	429 411 1 0	430 38 1 0
431 70 1 0	432 345 5 0	433 32 1 0	434 163 1 0	435 301 1 0
436 113 1 0	437 37 1 0	438 64 1 0	439 99 1 0	440 3 1 0
441 55 1 0	442 5 2 0	443 15 1 0	444 54 1 0	445 57 1 0
446 58 1 0	447 25 1 0	448 123 1 0	449 78 1 0	450 255 2 0
451 195 1 0	452 34 1 0	453 225 2 0	454 35 1 0	455 15 1 0
456 327 1 0	457 75 1 0	458 371 1 0	459 189 1 0	460 78 1 0
461 6 1 0	462 60 1 0	463 17 1 0	464 186 1 0	465 420 1 0
466 15 1 0	467 359 1 0	468 189 4 0	469 281 1 0	470 328 1 0
471 54 1 0	472 23 2 0	473 125 1 0	474 47 2 0	475 381 1 0
476 280 1 0	477 191 2 0	478 80 1 0	479 60 1 0	480 115 6 0
481 9 1 0 4	82 49 1 0	483 427 1 0	484 218 1 0	485 63 1 0
486 58 1 0	487 167 1 0	488 3 1 0	489 79 1 0	490 155 1 0
491 14 1 0	492 7 1 0	493 203 1 0	494 216 1 0	495 25 1 0
496 185 1 0	497 97 1 0	498 475 1 0	499 371 1 0	500 248 1 0
501 357 2 0	502 152 1 0	503 2 1 0	504 363 1 0	505 92 1 0
506 343 1 0	507 146 6 0	508 209 1 0	509 254 1 0	510 48 1 0
511 182 1 0	512 105 3 0	513 31 1 0	514 21 1 0	515 239 1 0
516 25 1 0	517 345 1 0	518 144 1 0	519 124 1 0	520 221 3 0
521 62 1 0	522 469 1 0	523 201 1 0	524 205 1 0	525 197 2 0
526 134 1 0	527 164 1 0	528 301 1 0	529 105 1 0	530 131 1 0
531 18 1 0	532 450 1 0	533 99 1 0	534 88 1 0	535 51 1 0
536 51 1 0	537 85 1 0	538 270 1 0	539 361 1 0	540 321 1 0
541 177 3 0	542 17 1 0	543 118 1 0	544 217 3 0	545 41 1 0
546 116 3 0	547 245 2 0	548 98 1 0	549 245 2 0	550 84 1 0
551 43 1 0	552 87 1 0	553 123 1 0	554 363 1 0	555 261 2 0
556 38 1 0	557 239 1 0	558 60 1 0	559 129 1 0	560 209 1 0
561 13 1 0	562 75 1 0	563 79 1 0	564 279 1 0	565 81 1 0
566 34 1 0	567 76 1 0	568 215 3 0	569 239 1 0	570 155 2 0
571 275 2 0	572 284 1 0	573 567 1 0	574 78 1 0	575 77 1 0
576 115 1 0	577 132 1 0	578 71 1 0	579 465 1 0	580 60 1 0
581 139 1 0	582 174 1 0	583 206 1 0	584 73 1 0	585 411 1 0
586 117 1 0	587 45 1 0	588 67 1 0	589 519 1 0	590 130 1 0
591 49 1 0	592 351 1 0	593 42 1 0	594 291 1 0	595 9 1 0
596 244 1 0	597 57 1 0	598 6 1 0	599 134 1 0	600 10 1 0
601 84 1 0	602 33 2 0	603 19 1 0	604 63 1 0	605 18 1 0
606 132 1 0	607 57 1 0	608 107 1 0	609 63 1 0	610 465 1 0
611 38 1 0	612 81 1 0	613 217 2 0	614 74 1 0	615 6 1 0
616 19 2 0	617 104 1 0	618 369 1 0	619 201 1 0	620 28 1 0
621 183 1 0	622 549 1 0	623 49 1 0	624 15 1 0	625 164 1 0
626 297 1 0	627 250 1 0	628 173 1 0	629 361 1 0	630 426 1 0
631 111 1 0	632 399 1 0	633 24 1 0	634 575 1 0	635 187 1 0
636 87 1 0	637 599 4 0	638 5 1 0	639 379 1 0	640 15 2 0
641 43 1 0	642 322 1 0	643 231 2 0	644 228 1 0	645 595 1 0
646 140 1 0	647 4 1 0	648 22 1 0	649 480 1 0	650 62 1 0
651 151 1 0	652 26 1 0	653 175 1 0	654 366 1 0	655 156 1 0
656 247 1 0	657 7 1 0	658 165 1 0	659 111 1 0	660 411 1 0
661 203 1 0	662 331 1 0	663 256 1 0	664 39 1 0	665 32 1 0
666 31 3 0	667 629 2 0	668 170 1 0	669 405 1 0	670 5 1 0
671 20 1 0	672 105 1 0	673 20 1 0	674 79 2 0	675 279 1 0
676 366 1 0	677 30 1 0	678 366 1 0	679 279 1 0	680 231 3 0
681 192 1 0	682 77 1 0	683 62 1 0	684 154 1 0	685 3 1 0
686 197 1 0	687 129 1 0	688 247 2 0	689 68 1 0	690 539 2 0
691 85 5 0	692 31 1 0	693 22 1 0	694 69 1 0	695 17 1 0
696 549 1 0	697 162 1 0	698 435 2 0	699 339 1 0	700 237 1 0
701 117 1 0	702 252 1 0	703 62 1 0	704 153 3 0	705 7 1 0
706 131 2 0	707 135 1 0	708 636 1 0	709 3 1 0	710 14 1 0
711 28 1 0	712 201 1 0	713 70 1 0	714 405 1 0	715 6 1 0
716 85 1 0	717 269 2 0	718 29 1 0	719 210 1 0	720 209 6 0
721 8 1 0	722 281 1 0	723 31 1 0	724 18 1 0	725 159 1 0
726 4 1 0	727 252 1 0	728 335 1 0	729 258 1 0	730 49 2 0
731 34 1 0	732 76 1 0	733 95 1 0	734 226 1 0	735 90 1 0
736 351 3 0	737 4 1 0	738 338 3 0	739 23 1 0	740 488 1 0
741 289 1 0	742 240 1 0	743 12 1 0	744 109 1 0	745 102 1 0
746 321 1 0	747 166 1 0	748 303 1 0	749 6 1 0	750 283 1 0
751 134 1 0	752 653 3 0	753 252 1 0	754 311 1 0	755 273 1 0
756 566 3 0	757 6 1 0	758 234 1 0	759 154 1 0	760 59 2 0
761 2 1 0	762 357 2 0	763 125 1 0	764 180 1 0	765 31 1 0
766 66 1 0	767 215 1 0	768 121 1 0	769 48 1 0	770 189 2 0
771 201 1 0	772 96 1 0	773 349 1 0	774 618 1 0	775 108 1 0
776 207 1 0	777 126 1 0	778 45 1 0	779 269 1 0	780 237 2 0
781 51 1 0	782 224 1 0	783 109 1 0	784 273 1 0	785 61 1 0
786 31 1 0	787 231 1 0	788 111 1 0	789 225 1 0	790 62 1 0
791 22 1 0	792 661 1 0	793 96 1 0	794 21 1 0	795 345 1 0
796 227 1 0	797 69 1 0	798 310 1 0	799 171 1 0	800 245 3 0

Note: "800 245 3 0" means $p(x) = (1 + x^3)(1 + x^{245}) + x^{800} = 1 + x^3 + x^{245} + x^{248} + x^{800}$.

Appendix 2: 9-Weight Primitive Polynomials of Degree up to 800 over GF(2)

==

				10 4 2 1 0
	12 6 4 1 0	13 9 2 1 0	14 9 2 1 0	15 7 2 1 0
16 9 3 1 0	17 8 2 1 0	18 5 2 1 0	19 10 2 1 0	20 9 2 1 0
21 5 2 1 0	22 6 2 1 0	23 11 2 1 0	24 6 2 1 0	25 4 2 1 0
26 7 3 1 0	27 6 2 1 0	28 15 2 1 0	29 14 2 1 0	30 6 2 1 0
31 4 2 1 0	32 14 3 1 0	33 10 2 1 0	34 21 2 1 0	35 14 2 1 0
36 5 2 1 0	37 15 2 1 0	38 18 2 1 0	39 21 2 1 0	40 7 2 1 0
41 20 2 1 0	42 12 2 1 0	43 7 3 1 0	44 9 2 1 0	45 21 2 1 0
46 22 2 1 0	47 9 2 1 0	48 22 3 1 0	49 15 2 1 0	50 35 2 1 0
51 6 2 1 0	52 16 2 1 0	53 5 2 1 0	54 7 2 1 0	55 31 2 1 0
56 13 3 1 0	57 18 2 1 0	58 16 2 1 0	59 5 2 1 0	60 36 2 1 0
61 9 2 1 0	62 11 2 1 0	63 15 2 1 0	64 6 2 1 0	65 23 2 1 0
66 5 2 1 0	67 15 3 1 0	68 38 2 1 0	69 6 2 1 0	70 19 3 1 0
71 24 2 1 0	72 6 2 1 0	73 36 2 1 0	74 11 2 1 0	75 49 2 1 0
76 27 2 1 0	77 17 2 1 0	78 69 2 1 0	79 16 2 1 0	80 17 3 1 0
81 40 2 1 0	82 28 2 1 0	83 53 2 1 0	84 30 2 1 0	85 31 2 1 0
86 9 3 1 0	87 9 2 1 0	88 59 3 1 0	89 26 2 1 0	90 42 2 1 0
91 9 2 1 0	92 26 2 1 0	93 41 2 1 0	94 21 2 1 0	95 8 2 1 0
96 75 2 1 0	97 18 2 1 0	98 8 2 1 0	99 7 3 1 0	100 39 2 1 0
101 15 2 1 0	102 25 2 1 0	103 31 2 1 0	104 17 3 1 0	105 22 2 1 0
106 21 2 1 0	107 7 3 1 0	108 51 2 1 0	109 18 2 1 0	110 53 2 1 0
111 18 2 1 0	112 7 3 1 0	113 38 2 1 0	114 17 2 1 0	115 61 2 1 0
116 66 2 1 0	117 17 2 1 0	118 6 2 1 0	119 20 2 1 0	120 15 2 1 0
121 24 2 1 0	122 12 2 1 0	123 5 3 1 0	124 61 2 1 0	125 23 2 1 0
126 6 2 1 0	127 4 2 1 0	128 119 2 1 0	129 36 2 1 0	130 42 2 1 0
131 5 2 1 0	132 18 2 1 0	133 51 2 1 0	134 45 2 1 0	135 8 2 1 0
136 7 2 1 0	137 32 2 1 0	138 12 2 1 0	139 18 2 1 0	140 69 2 1 0
141 5 2 1 0	142 22 2 1 0	143 15 2 1 0	144 15 3 1 0	145 30 2 1 0
146 49 3 1 0	147 22 2 1 0	148 57 2 1 0	149 57 2 1 0	150 63 2 1 0
151 12 2 1 0	152 9 3 1 0	153 52 2 1 0	154 34 2 1 0	155 14 3 1 0
156 42 2 1 0	157 109 2 1 0	158 26 2 1 0	159 38 2 1 0	160 46 2 1 0
161 68 2 1 0	162 40 2 1 0	163 106 2 1 0	164 5 2 1 0	165 61 2 1 0
166 7 2 1 0	167 24 2 1 0	168 27 4 1 0	169 33 2 1 0	170 72 2 1 0
171 9 2 1 0	172 4 2 1 0	173 9 3 1 0	174 86 3 1 0	175 13 2 1 0
176 93 4 1 0	177 65 2 1 0	178 127 3 1 0	179 25 3 1 0	180 37 2 1 0
181 51 2 1 0	182 51 2 1 0	183 70 2 1 0	184 35 3 1 0	185 29 2 1 0
186 17 2 1 0	187 14 3 1 0	188 57 3 1 0	189 10 2 1 0	190 151 2 1 0
191 24 2 1 0	192 125 3 1 0	193 45 2 1 0	194 13 3 1 0	195 101 2 1 0
196 36 2 1 0	197 5 2 1 0	198 30 2 1 0	199 36 2 1 0	200 171 2 1 0
201 94 2 1 0	202 18 2 1 0	203 20 3 1 0	204 127 2 1 0	205 151 2 1 0
206 102 2 1 0	207 21 2 1 0	208 25 3 1 0	209 95 2 1 0	210 91 2 1 0
211 141 2 1 0	212 105 2 1 0	213 9 2 1 0	214 7 3 1 0	215 72 2 1 0
216 49 3 1 0	217 94 2 1 0	218 8 2 1 0	219 25 2 1 0	220 38 3 1 0
221 75 2 1 0	222 17 2 1 0	223 46 2 1 0	224 53 3 1 0	225 48 2 1 0
226 81 2 1 0	227 5 2 1 0	228 173 2 1 0	229 87 2 1 0	230 84 3 1 0
231 20 2 1 0	232 29 3 1 0	233 14 2 1 0	234 102 2 1 0	235 42 2 1 0
236 72 2 1 0	237 6 2 1 0	238 7 2 1 0	239 50 2 1 0	240 30 3 1 0
241 21 2 1 0	242 165 2 1 0	243 14 2 1 0	244 51 2 1 0	245 98 2 1 0
246 155 2 1 0	247 69 2 1 0	248 75 2 1 0	249 33 2 1 0	250 208 2 1 0
251 57 3 1 0	252 14 2 1 0	253 57 2 1 0	254 197 2 1 0	255 36 2 1 0
256 145 3 1 0	257 23 2 1 0	258 86 2 1 0	259 61 2 1 0	260 35 2 1 0
261 55 2 1 0	262 22 2 1 0	263 38 2 1 0	264 7 2 1 0	265 70 2 1 0
266 72 2 1 0	267 17 2 1 0	268 141 2 1 0	269 75 2 1 0	270 10 3 1 0
271 67 2 1 0	272 5 3 1 0	273 35 2 1 0	274 55 2 1 0	275 106 3 1 0
276 65 2 1 0	277 43 2 1 0	278 135 2 1 0	279 17 2 1 0	280 247 2 1 0
281 41 2 1 0	282 66 2 1 0	283 57 2 1 0	284 162 2 1 0	285 38 2 1 0
286 18 2 1 0	287 143 2 1 0	288 15 2 1 0	289 76 2 1 0	290 50 2 1 0
291 11 3 1 0	292 178 2 1 0	293 18 2 1 0	294 141 2 1 0	295 10 2 1 0
296 251 2 1 0	297 12 2 1 0	298 24 2 1 0	299 105 3 1 0	300 39 2 1 0
301 133 2 1 0	302 207 2 1 0	303 66 2 1 0	304 135 2 1 0	305 33 2 1 0
306 30 2 1 0	307 189 2 1 0	308 30 2 1 0	309 21 2 1 0	310 31 2 1 0
311 14 2 1 0	312 121 3 1 0	313 30 2 1 0	314 185 2 1 0	315 70 2 1 0
316 112 2 1 0	317 99 2 1 0	318 14 2 1 0	319 46 2 1 0	320 147 2 1 0
321 76 2 1 0	322 15 2 1 0	323 6 2 1 0	324 48 2 1 0	325 39 2 1 0
326 113 2 1 0	327 59 2 1 0	328 30 2 1 0	329 84 2 1 0	330 56 2 1 0
331 237 2 1 0	332 98 2 1 0	333 87 2 1 0	334 201 2 1 0	335 23 2 1 0
336 107 2 1 0	337 21 2 1 0	338 203 2 1 0	339 12 3 1 0	340 241 2 1 0
341 75 2 1 0	342 283 2 1 0	343 27 2 1 0	344 27 2 1 0	345 53 2 1 0
346 40 2 1 0	347 17 2 1 0	348 129 2 1 0	349 10 2 1 0	350 29 2 1 0
351 17 2 1 0	352 31 2 1 0	353 134 2 1 0	354 43 2 1 0	355 121 3 1 0
356 62 2 1 0	357 145 2 1 0	358 105 2 1 0	359 72 2 1 0	360 251 2 1 0
361 12 2 1 0	362 42 2 1 0	363 65 2 1 0	364 13 2 1 0	365 28 3 1 0
366 73 2 1 0	367 55 2 1 0	368 271 3 1 0	369 102 2 1 0	370 184 2 1 0
371 257 2 1 0	372 62 2 1 0	373 67 2 1 0	374 8 3 1 0	375 109 2 1 0
376 75 2 1 0	377 18 2 1 0	378 23 2 1 0	379 30 3 1 0	380 15 2 1 0
381 179 2 1 0	382 231 2 1 0	383 131 2 1 0	384 31 2 1 0	385 84 2 1 0
386 20 2 1 0	387 249 2 1 0	388 175 2 1 0	389 194 2 1 0	390 95 2 1 0
391 21 2 1 0	392 17 6 1 0	393 26 2 1 0	394 31 2 1 0	395 29 3 1 0
396 54 2 1 0	397 115 2 1 0	398 8 3 1 0	399 13 2 1 0	400 81 3 1 0

Note: "12 6 4 1 0" means $p(x) = (1 + x)(1 + x^4)(1 + x^6) + x^{12} = 1 + x + x^4 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{12}$.

Appendix 2: 9-Weight Primitive Polynomials of Degree up to 800 over GF(2) – Cont'd

401 30 2 1 0	402 182 2 1 0	403 297 2 1 0	404 63 2 1 0	405 54 3 1 0
406 7 2 1 0	407 68 2 1 0	408 43 3 1 0	409 70 2 1 0	410 60 2 1 0
411 14 2 1 0	412 157 2 1 0	413 76 3 1 0	414 129 2 1 0	415 94 2 1 0
416 127 3 1 0	417 29 2 1 0	418 276 2 1 0	419 9 3 1 0	420 33 2 1 0
421 73 2 1 0	422 86 3 1 0	423 41 2 1 0	424 117 3 1 0	425 143 2 1 0
426 62 2 1 0	427 146 3 1 0	428 141 2 1 0	429 87 2 1 0	430 6 3 1 0
431 15 2 1 0	432 25 3 1 0	433 156 2 1 0	434 161 2 1 0	435 81 2 1 0
436 45 2 1 0	437 218 2 1 0	438 30 3 1 0	439 94 2 1 0	440 294 2 1 0
441 15 2 1 0	442 34 2 1 0	443 93 2 1 0	444 76 2 1 0	445 294 2 1 0
446 177 2 1 0	447 39 2 1 0	448 53 3 1 0	449 50 2 1 0	450 67 2 1 0
451 18 2 1 0	452 26 2 1 0	453 67 2 1 0	454 265 2 1 0	455 12 2 1 0
456 155 3 1 0	457 18 2 1 0	458 170 2 1 0	459 74 3 1 0	460 73 2 1 0
461 135 2 1 0	462 13 2 1 0	463 76 2 1 0	464 181 3 1 0	465 98 2 1 0
466 39 2 1 0	467 28 3 1 0	468 152 3 1 0	469 301 2 1 0	470 202 3 1 0
471 35 2 1 0	472 339 3 1 0	473 116 2 1 0	474 201 2 1 0	475 193 2 1 0
476 126 2 1 0	477 235 2 1 0	478 90 3 1 0	479 60 2 1 0	480 431 2 1 0
481 52 2 1 0	482 39 2 1 0	483 225 2 1 0	484 195 2 1 0	485 156 3 1 0
486 25 2 1 0	487 24 2 1 0	488 209 3 1 0	489 165 2 1 0	490 100 2 1 0
491 137 2 1 0	492 59 2 1 0	493 246 2 1 0	494 47 3 1 0	495 130 2 1 0
496 359 3 1 0	497 54 2 1 0	498 182 2 1 0	499 10 2 1 0	500 35 2 1 0
501 12 3 1 0	502 175 2 1 0	503 41 2 1 0	504 23 2 1 0	505 175 2 1 0
506 120 2 1 0	507 140 3 1 0	508 24 3 1 0	509 257 2 1 0	510 45 2 1 0
511 133 2 1 0	512 471 3 1 0	513 86 2 1 0	514 106 2 1 0	515 124 3 1 0
516 149 2 1 0	517 51 2 1 0	518 125 2 1 0	519 111 2 1 0	520 39 2 1 0
521 144 2 1 0	522 180 2 1 0	523 345 2 1 0	524 167 2 1 0	525 127 2 1 0
526 108 3 1 0	527 15 2 1 0	528 279 2 1 0	529 103 2 1 0	530 200 2 1 0
531 149 2 1 0	532 442 2 1 0	533 393 2 1 0	534 91 2 1 0	535 57 2 1 0
536 351 2 1 0	537 126 2 1 0	538 436 2 1 0	539 465 2 1 0	540 301 2 1 0
541 79 2 1 0	542 21 2 1 0	543 60 2 1 0	544 483 2 1 0	545 65 2 1 0
546 415 2 1 0	547 52 3 1 0	548 146 2 1 0	549 417 2 1 0	550 121 3 1 0
551 68 2 1 0	552 94 5 1 0	553 132 2 1 0	554 78 2 1 0	555 58 2 1 0
556 138 2 1 0	557 126 3 1 0	558 32 3 1 0	559 132 2 1 0	560 165 4 1 0
561 76 2 1 0	562 78 2 1 0	563 81 2 1 0	564 143 2 1 0	565 205 2 1 0
566 476 3 1 0	567 100 2 1 0	568 405 3 1 0	569 92 2 1 0	570 64 2 1 0
571 108 3 1 0	572 33 2 1 0	573 441 2 1 0	574 110 3 1 0	575 188 2 1 0
576 161 3 1 0	577 28 2 1 0	578 53 2 1 0	579 100 3 1 0	580 349 2 1 0
581 87 2 1 0	582 343 2 1 0	583 115 2 1 0	584 297 4 1 0	585 12 2 1 0
586 30 2 1 0	587 429 2 1 0	588 218 2 1 0	589 199 2 1 0	590 85 3 1 0
591 25 2 1 0	592 294 3 1 0	593 152 2 1 0	594 77 2 1 0	595 58 3 1 0
596 518 2 1 0	597 321 2 1 0	598 475 2 1 0	599 204 2 1 0	600 166 5 1 0
601 18 2 1 0	602 66 2 1 0	603 36 3 1 0	604 417 2 1 0	605 473 2 1 0
606 323 2 1 0	607 133 2 1 0	608 315 4 1 0	609 120 2 1 0	610 40 2 1 0
611 149 2 1 0	612 21 2 1 0	613 33 2 1 0	614 26 2 1 0	615 80 2 1 0
616 89 3 1 0	617 158 2 1 0	618 130 2 1 0	619 10 3 1 0	620 47 3 1 0
621 285 2 1 0	622 171 2 1 0	623 108 2 1 0	624 65 3 1 0	625 42 2 1 0
626 41 2 1 0	627 81 2 1 0	628 306 2 1 0	629 303 2 1 0	630 415 2 1 0
631 7 2 1 0	632 509 3 1 0	633 153 2 1 0	634 207 2 1 0	635 139 3 1 0
636 59 2 1 0	637 89 3 1 0	638 347 2 1 0	639 76 2 1 0	640 475 2 1 0
641 8 2 1 0	642 80 2 1 0	643 18 2 1 0	644 125 2 1 0	645 63 3 1 0
646 505 2 1 0	647 83 2 1 0	648 123 2 1 0	649 313 2 1 0	650 129 2 1 0
651 49 3 1 0	652 253 2 1 0	653 76 3 1 0	654 302 3 1 0	655 375 2 1 0
656 91 3 1 0	657 412 2 1 0	658 27 2 1 0	659 129 2 1 0	660 76 2 1 0
661 438 2 1 0	662 30 3 1 0	663 80 2 1 0	664 177 4 1 0	665 132 2 1 0
666 19 2 1 0	667 601 2 1 0	668 113 2 1 0	669 137 2 1 0	670 241 2 1 0
671 338 2 1 0	672 129 3 1 0	673 256 2 1 0	674 114 2 1 0	675 289 2 1 0
676 285 2 1 0	677 39 2 1 0	678 271 3 1 0	679 378 2 1 0	680 127 3 1 0
681 93 2 1 0	682 311 3 1 0	683 441 2 1 0	684 65 2 1 0	685 161 3 1 0
686 465 2 1 0	687 50 2 1 0	688 93 3 1 0	689 144 2 1 0	690 82 2 1 0
691 129 2 1 0	692 143 2 1 0	693 65 2 1 0	694 142 3 1 0	695 35 2 1 0
696 123 2 1 0	697 10 2 1 0	698 140 2 1 0	699 35 3 1 0	700 226 2 1 0
701 11 2 1 0	702 10 2 1 0	703 196 2 1 0	704 543 2 1 0	705 16 2 1 0
706 396 2 1 0	707 401 2 1 0	708 82 2 1 0	709 417 2 1 0	710 69 3 1 0
711 178 2 1 0	712 283 2 1 0	713 30 2 1 0	714 40 2 1 0	715 409 2 1 0
716 185 2 1 0	717 75 2 1 0	718 105 2 1 0	719 114 2 1 0	720 341 4 1 0
721 6 2 1 0	722 246 2 1 0	723 141 2 1 0	724 154 2 1 0	725 593 2 1 0
726 191 2 1 0	727 31 2 1 0	728 119 3 1 0	729 269 2 1 0	730 27 2 1 0
731 389 2 1 0	732 292 2 1 0	733 9 2 1 0	734 50 3 1 0	735 51 2 1 0
736 283 2 1 0	737 42 2 1 0	738 201 2 1 0	739 50 3 1 0	740 231 2 1 0
741 123 2 1 0	742 115 2 1 0	743 80 2 1 0	744 7 3 1 0	745 210 2 1 0
746 375 2 1 0	747 14 2 1 0	748 552 2 1 0	749 41 2 1 0	750 655 2 1 0
751 58 2 1 0	752 295 3 1 0	753 158 2 1 0	754 16 2 1 0	755 222 3 1 0
756 111 2 1 0	757 279 2 1 0	758 14 2 1 0	759 14 2 1 0	760 123 4 1 0
761 191 2 1 0	762 34 2 1 0	763 34 3 1 0	764 230 2 1 0	765 275 2 1 0
766 101 3 1 0	767 54 2 1 0	768 204 5 1 0	769 21 2 1 0	770 198 2 1 0
771 141 2 1 0	772 4 2 1 0	773 77 2 1 0	774 50 2 1 0	775 24 2 1 0
776 207 2 1 0	777 93 2 1 0	778 636 2 1 0	779 437 2 1 0	780 205 2 1 0
781 19 3 1 0	782 173 2 1 0	783 35 2 1 0	784 535 2 1 0	785 428 2 1 0
786 41 2 1 0	787 669 2 1 0	788 186 2 1 0	789 543 2 1 0	790 445 2 1 0
791 140 2 1 0	792 263 3 1 0	793 31 2 1 0	794 228 2 1 0	795 533 2 1 0
796 67 2 1 0	797 207 2 1 0	798 119 2 1 0	799 13 2 1 0	800 201 3 1 0

Note: "800 201 3 1 0" means $p(x) = (1 + x)(1 + x^3)(1 + x^{201}) + x^{800} = 1 + x + x^3 + x^4 + x^{201} + x^{202} + x^{204} + x^{205} + x^{800}$.

Appendix 3: 17-Weight Primitive Polynomials of Degree up to 800 over GF(2)

			19 9 4 2 1 0	20 10 6 2 1 0
21 9 4 2 1 0	22 9 4 2 1 0	23 14 4 2 1 0	24 10 4 2 1 0	25 14 4 2 1 0
26 15 5 2 1 0	27 18 5 2 1 0	28 9 4 2 1 0	29 13 4 2 1 0	30 9 4 2 1 0
31 11 4 2 1 0	32 15 4 2 1 0	33 12 4 2 1 0	34 24 4 2 1 0	35 10 4 2 1 0
36 15 4 2 1 0	37 9 4 2 1 0	38 10 4 2 1 0	39 21 4 2 1 0	40 18 5 2 1 0
41 26 4 2 1 0	42 21 4 2 1 0	43 19 5 2 1 0	44 22 4 2 1 0	45 31 4 2 1 0
46 23 4 2 1 0	47 14 4 2 1 0	48 27 4 2 1 0	49 8 4 2 1 0	50 15 5 2 1 0
51 32 5 2 1 0	52 21 5 2 1 0	53 22 4 2 1 0	54 16 6 2 1 0	55 45 4 2 1 0
56 18 5 2 1 0	57 28 4 2 1 0	58 11 4 2 1 0	59 33 4 2 1 0	60 30 4 2 1 0
61 32 5 2 1 0	62 30 5 2 1 0	63 15 4 2 1 0	64 39 4 2 1 0	65 21 4 2 1 0
66 48 4 2 1 0	67 45 4 2 1 0	68 9 4 2 1 0	69 26 5 2 1 0	70 39 4 2 1 0
71 11 4 2 1 0	72 41 5 2 1 0	73 11 4 2 1 0	74 29 4 2 1 0	75 20 5 2 1 0
76 17 4 2 1 0	77 9 4 2 1 0	78 43 4 2 1 0	79 15 4 2 1 0	80 9 5 2 1 0
81 28 4 2 1 0	82 21 4 2 1 0	83 61 5 2 1 0	84 33 4 2 1 0	85 21 4 2 1 0
86 21 5 2 1 0	87 10 4 2 1 0	88 27 5 2 1 0	89 20 4 2 1 0	90 36 4 2 1 0
91 69 4 2 1 0	92 25 4 2 1 0	93 46 4 2 1 0	94 11 4 2 1 0	95 42 4 2 1 0
96 75 5 2 1 0	97 20 4 2 1 0	98 12 4 2 1 0	99 63 5 2 1 0	100 30 4 2 1 0
101 46 4 2 1 0	102 63 5 2 1 0	103 24 4 2 1 0	104 42 6 2 1 0	105 22 4 2 1 0
106 90 4 2 1 0	107 16 5 2 1 0	108 78 5 2 1 0	109 18 4 2 1 0	110 18 6 2 1 0
111 19 4 2 1 0	112 37 6 2 1 0	113 8 4 2 1 0	114 17 5 2 1 0	115 28 5 2 1 0
116 11 4 2 1 0	117 27 6 2 1 0	118 11 5 2 1 0	119 29 4 2 1 0	120 93 5 2 1 0
121 54 4 2 1 0	122 19 4 2 1 0	123 73 4 2 1 0	124 52 5 2 1 0	125 31 4 2 1 0
126 55 4 2 1 0	127 8 4 2 1 0	128 63 4 2 1 0	129 45 4 2 1 0	130 44 4 2 1 0
131 82 4 2 1 0	132 75 4 2 1 0	133 15 4 2 1 0	134 95 4 2 1 0	135 12 4 2 1 0
136 53 5 2 1 0	137 18 4 2 1 0	138 61 4 2 1 0	139 77 4 2 1 0	140 55 4 2 1 0
141 109 4 2 1 0	142 65 5 2 1 0	143 33 4 2 1 0	144 99 4 2 1 0	145 45 4 2 1 0
146 27 4 2 1 0	147 46 4 2 1 0	148 45 5 2 1 0	149 65 4 2 1 0	150 49 4 2 1 0
151 8 4 2 1 0	152 14 4 2 1 0	153 61 4 2 1 0	154 53 4 2 1 0	155 28 5 2 1 0
156 84 4 2 1 0	157 111 4 2 1 0	158 15 4 2 1 0	159 28 4 2 1 0	160 49 5 2 1 0
161 20 4 2 1 0	162 81 4 2 1 0	163 65 4 2 1 0	164 62 4 2 1 0	165 30 5 2 1 0
166 9 4 2 1 0	167 50 4 2 1 0	168 58 6 2 1 0	169 15 4 2 1 0	170 16 4 2 1 0
171 65 5 2 1 0	172 28 5 2 1 0	173 107 4 2 1 0	174 145 4 2 1 0	175 17 4 2 1 0
176 63 4 2 1 0	177 75 4 2 1 0	178 132 4 2 1 0	179 15 5 2 1 0	180 97 4 2 1 0
181 63 4 2 1 0	182 89 4 2 1 0	183 21 4 2 1 0	184 87 5 2 1 0	185 101 4 2 1 0
186 58 4 2 1 0	187 129 4 2 1 0	188 133 4 2 1 0	189 54 4 2 1 0	190 177 5 2 1 0
191 23 4 2 1 0	192 117 7 2 1 0	193 8 4 2 1 0	194 52 4 2 1 0	195 117 4 2 1 0
196 80 4 2 1 0	197 18 4 2 1 0	198 9 4 2 1 0	199 48 4 2 1 0	200 107 4 2 1 0
201 21 4 2 1 0	202 66 4 2 1 0	203 130 4 2 1 0	204 36 4 2 1 0	205 29 4 2 1 0
206 41 4 2 1 0	207 115 4 2 1 0	208 35 5 2 1 0	209 21 4 2 1 0	210 156 4 2 1 0
211 189 4 2 1 0	212 23 4 2 1 0	213 13 4 2 1 0	214 119 4 2 1 0	215 16 4 2 1 0
216 47 5 2 1 0	217 50 4 2 1 0	218 63 4 2 1 0	219 69 5 2 1 0	220 24 4 2 1 0
221 95 4 2 1 0	222 47 6 2 1 0	223 38 4 2 1 0	224 203 4 2 1 0	225 42 4 2 1 0
226 68 4 2 1 0	227 165 4 2 1 0	228 36 4 2 1 0	229 83 4 2 1 0	230 141 4 2 1 0
231 43 4 2 1 0	232 21 5 2 1 0	233 22 4 2 1 0	234 52 4 2 1 0	235 141 4 2 1 0
236 17 4 2 1 0	237 13 4 2 1 0	238 177 4 2 1 0	239 69 4 2 1 0	240 16 6 2 1 0
241 21 4 2 1 0	242 23 4 2 1 0	243 105 5 2 1 0	244 159 4 2 1 0	245 31 4 2 1 0
246 49 4 2 1 0	247 30 4 2 1 0	248 215 6 2 1 0	249 18 4 2 1 0	250 90 4 2 1 0
251 133 4 2 1 0	252 174 5 2 1 0	253 189 4 2 1 0	254 67 4 2 1 0	255 61 4 2 1 0
256 245 5 2 1 0	257 170 4 2 1 0	258 24 4 2 1 0	259 9 4 2 1 0	260 49 4 2 1 0
261 33 4 2 1 0	262 33 4 2 1 0	263 21 4 2 1 0	264 123 5 2 1 0	265 110 4 2 1 0
266 25 4 2 1 0	267 121 4 2 1 0	268 131 4 2 1 0	269 49 5 2 1 0	270 189 4 2 1 0
271 24 4 2 1 0	272 243 4 2 1 0	273 16 4 2 1 0	274 60 4 2 1 0	275 39 5 2 1 0
276 103 4 2 1 0	277 45 4 2 1 0	278 147 4 2 1 0	279 33 4 2 1 0	280 14 4 2 1 0
281 67 4 2 1 0	282 25 4 2 1 0	283 110 4 2 1 0	284 14 4 2 1 0	285 105 4 2 1 0
286 15 4 2 1 0	287 20 4 2 1 0	288 65 5 2 1 0	289 68 4 2 1 0	290 26 4 2 1 0
291 45 6 2 1 0	292 80 4 2 1 0	293 187 4 2 1 0	294 31 4 2 1 0	295 14 4 2 1 0
296 203 4 2 1 0	297 144 4 2 1 0	298 38 4 2 1 0	299 21 5 2 1 0	300 246 4 2 1 0
301 9 4 2 1 0	302 63 4 2 1 0	303 15 4 2 1 0	304 255 4 2 1 0	305 112 4 2 1 0
306 156 4 2 1 0	307 185 4 2 1 0	308 124 4 2 1 0	309 99 4 2 1 0	310 81 5 2 1 0
311 59 4 2 1 0	312 29 6 2 1 0	313 14 4 2 1 0	314 38 4 2 1 0	315 117 5 2 1 0
316 210 4 2 1 0	317 275 4 2 1 0	318 171 4 2 1 0	319 144 4 2 1 0	320 139 6 2 1 0
321 15 4 2 1 0	322 113 4 2 1 0	323 105 4 2 1 0	324 18 4 2 1 0	325 50 5 2 1 0
326 58 4 2 1 0	327 145 4 2 1 0	328 71 5 2 1 0	329 50 4 2 1 0	330 172 4 2 1 0
331 19 5 2 1 0	332 314 4 2 1 0	333 57 4 2 1 0	334 189 4 2 1 0	335 22 4 2 1 0
336 209 5 2 1 0	337 128 4 2 1 0	338 186 4 2 1 0	339 13 4 2 1 0	340 186 4 2 1 0
341 62 4 2 1 0	342 81 5 2 1 0	343 36 4 2 1 0	344 27 4 2 1 0	345 112 4 2 1 0
346 150 4 2 1 0	347 15 5 2 1 0	348 105 4 2 1 0	349 83 4 2 1 0	350 165 4 2 1 0
351 88 4 2 1 0	352 99 4 2 1 0	353 55 4 2 1 0	354 126 4 2 1 0	355 38 5 2 1 0
356 63 4 2 1 0	357 325 4 2 1 0	358 65 4 2 1 0	359 43 4 2 1 0	360 162 7 2 1 0
361 50 4 2 1 0	362 308 4 2 1 0	363 33 4 2 1 0	364 63 4 2 1 0	365 197 4 2 1 0
366 9 4 2 1 0	367 53 4 2 1 0	368 87 4 2 1 0	369 24 4 2 1 0	370 174 4 2 1 0
371 22 4 2 1 0	372 268 4 2 1 0	373 77 4 2 1 0	374 67 4 2 1 0	375 91 4 2 1 0
376 29 5 2 1 0	377 45 4 2 1 0	378 120 4 2 1 0	379 357 4 2 1 0	380 95 4 2 1 0
381 159 4 2 1 0	382 69 5 2 1 0	383 83 4 2 1 0	384 365 5 2 1 0	385 15 4 2 1 0
386 14 4 2 1 0	387 261 4 2 1 0	388 111 4 2 1 0	389 57 4 2 1 0	390 99 6 2 1 0
391 38 4 2 1 0	392 163 6 2 1 0	393 58 4 2 1 0	394 14 4 2 1 0	395 10 5 2 1 0
396 178 4 2 1 0	397 35 4 2 1 0	398 114 5 2 1 0	399 31 4 2 1 0	400 31 6 2 1 0

Note: "20 10 6 2 1 0" means $p(x) = (1 + x)(1 + x^2)(1 + x^6)(1 + x^{10}) + x^{20}$.

Appendix 3: 17-Weight Primitive Polynomials of Degree up to 800 over GF(2) – Cont'd

401 91 4 2 1 0	402 133 4 2 1 0	403 389 4 2 1 0	404 38 4 2 1 0	405 23 5 2 1 0
406 315 4 2 1 0	407 41 4 2 1 0	408 155 6 2 1 0	409 150 4 2 1 0	410 60 4 2 1 0
411 229 4 2 1 0	412 341 4 2 1 0	413 54 4 2 1 0	414 141 4 2 1 0	415 219 4 2 1 0
416 381 5 2 1 0	417 120 4 2 1 0	418 222 4 2 1 0	419 301 5 2 1 0	420 34 4 2 1 0
421 237 4 2 1 0	422 277 5 2 1 0	423 108 4 2 1 0	424 9 5 2 1 0	425 20 4 2 1 0
426 138 4 2 1 0	427 113 4 2 1 0	428 28 4 2 1 0	429 63 4 2 1 0	430 142 5 2 1 0
431 268 4 2 1 0	432 62 5 2 1 0	433 90 4 2 1 0	434 13 4 2 1 0	435 261 4 2 1 0
436 138 4 2 1 0	437 25 4 2 1 0	438 100 6 2 1 0	439 21 4 2 1 0	440 205 5 2 1 0
441 48 4 2 1 0	442 21 4 2 1 0	443 39 5 2 1 0	444 126 4 2 1 0	445 99 4 2 1 0
446 409 4 2 1 0	447 61 4 2 1 0	448 137 5 2 1 0	449 104 4 2 1 0	450 225 4 2 1 0
451 32 5 2 1 0	452 365 4 2 1 0	453 259 4 2 1 0	454 33 4 2 1 0	455 80 4 2 1 0
456 203 5 2 1 0	457 210 4 2 1 0	458 213 4 2 1 0	459 257 5 2 1 0	460 35 5 2 1 0
461 71 4 2 1 0	462 9 5 2 1 0	463 14 4 2 1 0	464 68 6 2 1 0	465 61 4 2 1 0
466 41 5 2 1 0	467 54 5 2 1 0	468 300 4 2 1 0	469 57 4 2 1 0	470 379 5 2 1 0
471 19 4 2 1 0	472 229 6 2 1 0	473 88 4 2 1 0	474 121 4 2 1 0	475 143 5 2 1 0
476 26 4 2 1 0	477 197 5 2 1 0	478 177 4 2 1 0	479 63 4 2 1 0	480 166 7 2 1 0
481 266 4 2 1 0	482 180 4 2 1 0	483 322 4 2 1 0	484 23 4 2 1 0	485 205 4 2 1 0
486 97 4 2 1 0	487 78 4 2 1 0	488 14 4 2 1 0	489 22 4 2 1 0	490 47 4 2 1 0
491 329 4 2 1 0	492 460 4 2 1 0	493 18 5 2 1 0	494 30 4 2 1 0	495 156 4 2 1 0
496 399 4 2 1 0	497 94 4 2 1 0	498 69 4 2 1 0	499 10 5 2 1 0	500 82 4 2 1 0
501 152 5 2 1 0	502 93 4 2 1 0	503 119 4 2 1 0	504 459 4 2 1 0	505 104 4 2 1 0
506 203 4 2 1 0	507 369 4 2 1 0	508 162 4 2 1 0	509 35 4 2 1 0	510 397 4 2 1 0
511 8 4 2 1 0	512 63 4 2 1 0	513 30 4 2 1 0	514 158 4 2 1 0	515 61 5 2 1 0
516 229 4 2 1 0	517 15 4 2 1 0	518 271 4 2 1 0	519 114 4 2 1 0	520 81 5 2 1 0
521 125 4 2 1 0	522 300 5 2 1 0	523 15 5 2 1 0	524 21 4 2 1 0	525 207 4 2 1 0
526 61 5 2 1 0	527 263 4 2 1 0	528 233 5 2 1 0	529 18 4 2 1 0	530 65 4 2 1 0
531 45 4 2 1 0	532 281 4 2 1 0	533 117 5 2 1 0	534 325 6 2 1 0	535 21 4 2 1 0
536 403 4 2 1 0	537 258 5 2 1 0	538 42 4 2 1 0	539 40 5 2 1 0	540 154 4 2 1 0
541 189 4 2 1 0	542 249 5 2 1 0	543 31 4 2 1 0	544 311 4 2 1 0	545 321 4 2 1 0
546 72 4 2 1 0	547 26 5 2 1 0	548 43 4 2 1 0	549 195 4 2 1 0	550 95 4 2 1 0
551 23 4 2 1 0	552 441 5 2 1 0	553 32 4 2 1 0	554 53 4 2 1 0	555 49 4 2 1 0
556 336 4 2 1 0	557 18 4 2 1 0	558 427 4 2 1 0	559 15 4 2 1 0	560 167 4 2 1 0
561 21 4 2 1 0	562 84 4 2 1 0	563 81 4 2 1 0	564 478 4 2 1 0	565 160 5 2 1 0
566 73 5 2 1 0	567 88 4 2 1 0	568 109 6 2 1 0	569 11 4 2 1 0	570 217 4 2 1 0
571 107 5 2 1 0	572 229 4 2 1 0	573 127 4 2 1 0	574 89 4 2 1 0	575 62 4 2 1 0
576 499 4 2 1 0	577 90 4 2 1 0	578 144 4 2 1 0	579 541 4 2 1 0	580 69 4 2 1 0
581 14 4 2 1 0	582 129 5 2 1 0	583 200 4 2 1 0	584 121 5 2 1 0	585 190 4 2 1 0
586 78 5 2 1 0	587 37 5 2 1 0	588 12 4 2 1 0	589 519 4 2 1 0	590 394 4 2 1 0
591 544 4 2 1 0	592 145 5 2 1 0	593 14 4 2 1 0	594 232 4 2 1 0	595 380 5 2 1 0
596 170 4 2 1 0	597 293 5 2 1 0	598 105 4 2 1 0	599 63 4 2 1 0	600 327 4 2 1 0
601 62 4 2 1 0	602 58 4 2 1 0	603 21 4 2 1 0	604 441 4 2 1 0	605 111 4 2 1 0
606 105 4 2 1 0	607 56 4 2 1 0	608 199 4 2 1 0	609 190 4 2 1 0	610 50 4 2 1 0
611 165 4 2 1 0	612 181 4 2 1 0	613 401 4 2 1 0	614 143 4 2 1 0	615 42 5 2 1 0
616 56 6 2 1 0	617 153 4 2 1 0	618 182 5 2 1 0	619 293 4 2 1 0	620 66 4 2 1 0
621 223 4 2 1 0	622 445 5 2 1 0	623 70 4 2 1 0	624 161 5 2 1 0	625 330 4 2 1 0
626 307 4 2 1 0	627 440 5 2 1 0	628 51 4 2 1 0	629 215 4 2 1 0	630 207 4 2 1 0
631 71 4 2 1 0	632 165 5 2 1 0	633 229 4 2 1 0	634 570 4 2 1 0	635 241 4 2 1 0
636 219 5 2 1 0	637 155 4 2 1 0	638 141 4 2 1 0	639 355 4 2 1 0	640 199 5 2 1 0
641 29 4 2 1 0	642 360 4 2 1 0	643 377 4 2 1 0	644 60 4 2 1 0	645 15 4 2 1 0
646 301 5 2 1 0	647 143 4 2 1 0	648 33 5 2 1 0	649 75 4 2 1 0	650 197 4 2 1 0
651 477 4 2 1 0	652 329 4 2 1 0	653 77 4 2 1 0	654 277 4 2 1 0	655 200 4 2 1 0
656 147 5 2 1 0	657 96 4 2 1 0	658 162 4 2 1 0	659 361 4 2 1 0	660 196 4 2 1 0
661 21 4 2 1 0	662 369 5 2 1 0	663 54 4 2 1 0	664 527 5 2 1 0	665 101 4 2 1 0
666 90 4 2 1 0	667 245 4 2 1 0	668 66 4 2 1 0	669 169 4 2 1 0	670 227 4 2 1 0
671 8 4 2 1 0	672 19 6 2 1 0	673 90 4 2 1 0	674 96 4 2 1 0	675 61 4 2 1 0
676 170 4 2 1 0	677 443 4 2 1 0	678 657 4 2 1 0	679 18 4 2 1 0	680 399 5 2 1 0
681 280 4 2 1 0	682 363 4 2 1 0	683 100 5 2 1 0	684 510 4 2 1 0	685 37 5 2 1 0
686 477 4 2 1 0	687 135 4 2 1 0	688 89 5 2 1 0	689 45 4 2 1 0	690 186 4 2 1 0
691 381 4 2 1 0	692 194 4 2 1 0	693 279 4 2 1 0	694 141 4 2 1 0	695 61 4 2 1 0
696 315 4 2 1 0	697 228 4 2 1 0	698 145 4 2 1 0	699 453 4 2 1 0	700 95 4 2 1 0
701 169 4 2 1 0	702 93 5 2 1 0	703 195 4 2 1 0	704 208 6 2 1 0	705 154 4 2 1 0
706 63 4 2 1 0	707 202 5 2 1 0	708 110 5 2 1 0	709 81 4 2 1 0	710 31 4 2 1 0
711 186 4 2 1 0	712 383 4 2 1 0	713 14 4 2 1 0	714 16 4 2 1 0	715 63 5 2 1 0
716 61 4 2 1 0	717 231 5 2 1 0	718 99 4 2 1 0	719 24 4 2 1 0	720 389 5 2 1 0
721 141 4 2 1 0	722 50 4 2 1 0	723 93 4 2 1 0	724 116 4 2 1 0	725 79 4 2 1 0
726 9 4 2 1 0	727 41 4 2 1 0	728 571 4 2 1 0	729 153 4 2 1 0	730 480 4 2 1 0
731 329 4 2 1 0	732 126 4 2 1 0	733 83 4 2 1 0	734 133 4 2 1 0	735 36 4 2 1 0
736 143 7 2 1 0	737 25 4 2 1 0	738 13 4 2 1 0	739 55 5 2 1 0	740 705 4 2 1 0
741 553 4 2 1 0	742 93 4 2 1 0	743 49 4 2 1 0	744 167 5 2 1 0	745 677 4 2 1 0
746 87 4 2 1 0	747 13 4 2 1 0	748 104 4 2 1 0	749 345 4 2 1 0	750 129 5 2 1 0
751 54 4 2 1 0	752 206 6 2 1 0	753 336 4 2 1 0	754 26 4 2 1 0	755 114 5 2 1 0
756 208 4 2 1 0	757 141 4 2 1 0	758 637 4 2 1 0	759 40 4 2 1 0	760 231 5 2 1 0
761 467 4 2 1 0	762 600 4 2 1 0	763 14 4 2 1 0	764 413 4 2 1 0	765 438 5 2 1 0
766 161 4 2 1 0	767 423 4 2 1 0	768 209 6 2 1 0	769 119 4 2 1 0	770 736 4 2 1 0
771 513 4 2 1 0	772 126 4 2 1 0	773 273 4 2 1 0	774 15 4 2 1 0	775 84 4 2 1 0
776 631 4 2 1 0	777 160 4 2 1 0	778 199 5 2 1 0	779 51 5 2 1 0	780 307 4 2 1 0
781 725 4 2 1 0	782 385 4 2 1 0	783 64 4 2 1 0	784 207 4 2 1 0	785 364 4 2 1 0
786 114 4 2 1 0	787 177 5 2 1 0	788 25 4 2 1 0	789 399 4 2 1 0	790 545 4 2 1 0
791 35 4 2 1 0	792 737 5 2 1 0	793 110 4 2 1 0	794 152 4 2 1 0	795 37 4 2 1 0
796 9 4 2 1 0	797 415 5 2 1 0	798 765 4 2 1 0	799 72 4 2 1 0	800 512 6 2 1 0

Note: "800 512 6 2 1 0" means $p(x) = (1 + x)(1 + x^2)(1 + x^6)(1 + x^{512}) + x^{800}$.