NLFSR Functions with Optimal Periods

Sultan Almuhammadi, Ibraheem Al-Hejri,
Ghashmi Bin Talib, and Awadh Gaamel

King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
muhamadi@kfupm.edu.sa, alhejri87@gmail.com,
(g201309530, g201402020)@kfupm.edu.sa

Abstract. Nonlinear feedback shift registers (NLFSRs) are basic components, typically found in stream ciphers and other cryptosystems. The main purpose of these components is to generate pseudorandom sequences of bits. There is no mathematical foundation on how to construct an NLFSR feedback function with optimal period. In this work, we review the existing NLFSR feedback functions, and propose new functions with optimal periods.

Keywords: NLFSR, pseudorandom, optimal period

1 Introduction

One of the most prevalent tools used to generate pseudo-random sequences is the feedback shift register (FSR). FSRs have various applications such as data compression [1], cryptography [2], error detection and correction [3] and testing [4]. Research on LFSRs has been developed since early 1960's [5].

Depending on its internal feedback function, the FSR can be either linear (LFSR) or nonlinear (NLFSR). The area of LFSRs is considered well-known and most of the LFSR fundamental problems are solved. A designer only has to use a primitive generator polynomial in order to build a LFSR of size n bits with a maximum period.

Unlike LFSR, the state in NLFSR is a non-linear transformation of the previous state [6]. The importance of NLFSR comes from its ability to produce a very secure pseudorandom sequence which is hard to break. Thus, to determine the structure of an NLFSR, at least $\Theta(2^n)$ sequence bits are needed, where n is the register size [7], while only $2n$ bits are needed to determine the structure of an LFSR. However, NLFSRs still have many fundamental problems, which remain open even with well-known theory. A systematic method to construct an NLFSR with optimal periods is the most significant problem to be solved. This problem remains challenging since there is no mathematical foundation supporting it.

In this paper, we propose new feedback functions with optimal periods for NLFSRs by construction. The rest of the paper is organized as follows. Section 2 presents the concept of feedback shift registers and related definitions. Section 3 reviews the existing NLFSR feedback functions which have optimal periods. Section 4 describes our construction method. Section 5 presents the new feedback functions. Finally, Section 6 concludes and outlines future work.

2 Preliminaries

An FSR consists of a shift-register to store the state, and a function to compute the feedback bit. The register has n binary storage cells, each cell $i \in \{0, 1, \ldots, n-1\}$ holds a single bit x_i in the state register. The feedback function $f : \{0,1\}^n \to \{0,1\}$ computes a feedback bit used as an input to the register to update the state using a shift operation. Fig. 1 illustrates an n-bit FSR general structure.

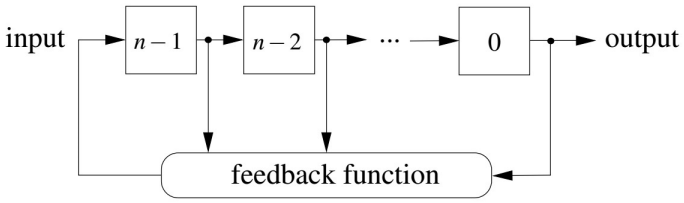


Fig. 1: An n-bit FSR general structure

The state of an FSR is presented by a vector of bits $X = (x_0, x_1, \ldots, x_{n-1})$. An FSR generates a pseudorandom sequence of bits obtained by extracting the bits from the shift-register one-by-one. The initial state X_0 is a seed used to generate the output sequence of bits. From the current state, the feedback function computes x_n, which is the input to the FSR and it updates the input cell $(n-1)$, while the value x_0 of the cell 0 determines the output of the FSR.

Due to the limitation of register size, n, there will be a repeated state eventually (when all the 2^n states are exhausted). The *period* of the FSR is defined as the length of the longest unrepeated output sequence. Let us now proceed formally.

Definition 1: Let $\mathbb{F}_2$ be the binary finite field and $\mathbb{F}_2[x]$ represent the ring of polynomials in undefined value of x with coefficients taken from $\mathbb{F}_2$. Assume that, $\mathbb{F}_2^n$ is an n-dimensional vector space over $\mathbb{F}_2$ which consists of n-tuples of $\mathbb{F}_2$ elements. Each function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is represented by a Boolean function of n variables. The elements sequence $s = (s_0, s_1, \ldots)$ of $\mathbb{F}_2$ is known as a binary sequence. The sequence $s = (s_i)_{i=0}^{\infty}$ is *periodic* when there is a positive integer w such that $s_{i+w} = s_i, \forall i \geq 0$. The least such positive integer is called a *period*.

In a binary n-stage FSR, $\mathbb{F}_2^n$ is mapped into $\mathbb{F}_2^n$ as shown below, where F is the mapping function:

$$F(x_0, x_1, \ldots, x_{n-1}) = ((x_1, \ldots, x_{n-1}, f(x_0, x_1, \ldots, x_{n-1}))$$

The feedback function is the Boolean function f over n-variables. If F is a linear transformation from $\mathbb{F}_2^n$ to itself, it is called a *linear feedback shift register* (LFSR), otherwise it is called a *nonlinear feedback shift register* (NLFSR). Moreover, if the function F is mapped as a bijection, it is called *non-singular* [8].

LFSRs have feedback functions of the following type:

$$f(x_0, x_1, \ldots, x_{n-1}) = c_0 \cdot x_0 \oplus c_1 \cdot x_1 \oplus \ldots c_{n-1} \cdot x_{n-1}$$

where $c_i \in \{0, 1\}$ for $i \in \{0, 1, \ldots, n-1\}$). All terms in an LFSR are linear, like $(c_i \cdot x_i)$. However, NLFSRs include nonlinear terms. For example, an NLFSR of degree 2 includes at least one term of the form $(x_i \cdot x_j)$, where $i \neq j$.

Definition 2: The sequence of de Bruijn $(a_0, \ldots, a_{2^n-1})$ of order n which consists of elements from $\mathbb{F}_2$ has a period of 2^n where each n-tuples appears exactly once.

Sainte-Marie [9] and de Bruijn [10,11] show that the number of sequences which are cyclic equivalent is given by:

$$B_n = 2^{2^{n-1}-n} \tag{1}$$

Definition 3: The modified version of de Bruijn sequence $(a_0, ..., a_{2^n-2})$ of order n is a sequence which has a period of 2^{n-1}.

In FSR, the all-zeroes state ($X = 0$) should not be allowed at any given point, or otherwise the FSR will remain locked up at this state. If $X = 0$, then $f(X) = 0$ for all new states. Hence, the sequence s will be all zeros. Therefore, the maximum period of an FSR of size n is at most $2^n - 1$. It is important to include the output bit x_0 in the feedback function to maximize the period. Hence, the FSR can take the following form:

$$f(x_0, x_1, \ldots, x_{n-1}) = x_0 + g(x_1, x_2, \ldots, x_{n-1}) \tag{2}$$

Where g denotes to a Boolean function on $n-1$ variables.

Mayhew and Golomb [11] investigated sequences satisfying Definition 3. Gammel et al. [12] called these sequences primitive. In LFSRs, these sequences can be produced using primitive polynomials and the theory of such sequences is well-understood [13].

The primitive sequence is a significant factor in the applications of cryptography. The number of primitive sequences jointly (in linear and nonlinear functions) is given by B_n in (1). We have $\phi(2^{n-1})/n$ primitive LFSRs, where ϕ refers to the Euler phi function. Since we have $2^{2^{n-1}}$ Boolean functions on $n-1$ variables, the probability of selecting a primitive NLFSR function of the form in (2) is given by:

$$\frac{(2^{2^{n-1}-n})}{2^{2^{n-1}}} = \frac{1}{2^n} \tag{3}$$

Thus, there are more NLFSRs than LFSRs [8].

Berlekamp-Massey algorithm [14] can be used to generate a given binary sequence of a minimal LFSR. Golomb's postulates [5] have characterized the properties of sequences created by LFSRs statistically.

In LFSR system, there is a unique transformation between the configurations of Galois and Fibonacci. Therefore, we can reverse the order of LFSR's feedback taps and adjust the initial state to get the configuration of Galois from Fibonacci (or vice versa). In contrast, the NLFSR system does not have a unique transformation between Fibonacci and Galois configurations [15,16,17].

3 Literature Review

Many researchers focused on non-linear feedback shift register functions with optimal periods. However, much of their work is either restricted to particular cases [18,19], or limited to a subset of functions for particular values of n (like $n = 1$, 3, 4, 11-13, 15-17, 19-21, 23, 31-33) [20]. Generally, the NLFSR problem is very hard due to the lack of the mathematical foundation on how to construct the feedback functions that give optimal periods.

An NLFSR can be implemented using either Fibonacci or Galois configuration. The Fibonacci configuration applies the feedback only to the input cell $(n-1)$ of the shift register as shown in Fig. 1, while the Galois configuration is able to potentially apply the feedback to any cell. Fig. 2 shows a 4-bit Fibonacci NLFSR with its feedback function $f(x_0, x_1, x_2, x_3) = x_0 \oplus x_1 \oplus x_2 \oplus x_1 \cdot x_2$.

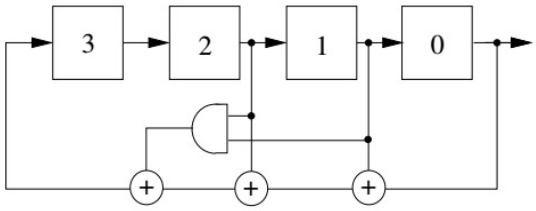


Fig. 2: A 4-bit Fibonacci NLFSR example. [21]

Dubrova [21] presented Fibonacci NLFSR feedback functions of degree 2, with optimal periods of $2^n - 1$, for $4 \leq n \leq 25$, in the following three types:

- Type 1: $f(x_0, x_1, \ldots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \cdot x_d$
- Type 2: $f(x_0, x_1, \ldots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \cdot x_c \oplus x_d \cdot x_e$
- Type 3: $f(x_0, x_1, \ldots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \oplus x_d \oplus x_e \cdot x_h$

where $a, b, c, d, e, h \in \{1, 2, \ldots, n-1\}$, $x_i \in \{0, 1\}$, and the addition (XOR) and multiplication (AND) operations are in modulo 2.

Mandal and Gong [22] have defined the relation between an NLFSR and a regular directed graph on a field extension. They assumed two states k^{th} and $(k+1)^{st}$ such that $S_k = (a_k, a_{k+1}, \ldots, a_{k+n-1})$ and $S_{k+1} = (a_{k+1}, a_{k+2}, \ldots, a_{k+n})$, where $S_{k+1} = f(S_k)$, for $k \geq 0$ and f refers to the feedback function. They used $G = (V, E)$ to be the directed graph defined as follows:

- Each state S_k is denoted as a vertex $v_k \in V$.
- There is a directed edge $e_k \in E$ between the state S_k to the state S^i_{k+1}.
- $S^i_{k+1} = (a_{k+1}, a_{k+2}, \ldots, a_{n+k-1}, a^i_{n+k})$, where $a^i_{n+k} \neq a^j_{n+k}$, $i \neq j$, for $i = 1, 2, 3, \ldots, 2^q$.

The directed graph (G) here is called de Bruijn graph [10,23], which consists of 2^q regular graphs along with $|V| = 2^{qn}$ and $|E| = 2^{q(n+1)}$. They relied on a uniform probability distribution Ω to define the random feedback functions, as:

$$F = (f, \Omega)$$

where the uniform probability is computed by $p_j = \frac{1}{2^q}$, for $j = 1, 2, \ldots, 2^q$. The input state is S_k and the output of F is $S^i_{k+1}, (F(S_k) = S^i_{k+1})$ for some values of i which is selected depending on f and Ω where S^i_{k+1} is not generated by F. Let S_0 be an initial state, then the random NLFSR sequence, that F can generate, is given by $a = \{a_0, a_1, \ldots, a_k, a_{k+1}, \ldots\}$ and the sequence period is P, for $S_0 = F^P(S_0)$, where $F^P(S) = F^{P-1}(F(S))$.

Mandal and Gong used a random walk definition on the directed graph G. They used it among the vertices v_k and v_{k+1} imposing the distribution of uniform probability defined as follows:

- Assume S_k and S_{k+1} are the states which are corresponding to the vertices v_k and v_{k+1}.
- Next, at v_k, the random walk selects randomly v_{k+1} according to F beneath uniform distribution $Pr(S_{k+1} = F(S_k)) = \frac{1}{2^q}$.

Motwan et al. [24] and Banderier et al. [25] considered a basic random walk R(P) on the directed graph G of length P which works as follows:

- Start from any vertex v_0.
- The next vertex v_1 will be selected from v_0 with probability $Pr(S_1 = F(S_0)) = \frac{1}{2^q}$, if v_1 has not been visited yet.
- Repeat the second step until it finds v_0.
- If v_0 has been found, stop.

Therefore, the sequence $\{a_i\}$ generated by F and the random walk are equivalent. Obtaining the value of P in $R(P)$ means obtaining the NLFSR sequence period [25].

4 Constructing NLFSR Feedback Functions

In this work, we propose and use an efficient sequential NLFSR function construction method, which sequentially enumerates all feedback functions of a desired type. We then compute the period for each feedback function and verify its optimality. The normal straightforward method to check all values located in a register is very expensive in terms of both time and computational power since the number of functions grows exponentially in terms of n.

4.1 Proposed Construction Method

The proposed construction method consists of two parts:

1. *Sequential Function Generator (SFG)*: which is an enumeration of the feedback functions. Thus, given a specific type of a function f and its size n, the SFG sequentially increments the indices of the variables $(x_a, x_b, x_c, \dots)$ in f such that all nonequivalent functions are enumerated. Since x_0 is a fixed input in all the three types of feedback functions, all other indices are chosen relatively to x_0 such that the constructed functions are nonequivalent. For the linear terms of f, the first index a is initially assigned to 1, while the next index b (in types 1 and 3) is assigned to $a+1$ and sequentially incremented. When b reaches to the end, the previous index a is incremented and b is reset to $a+1$ again, and so on. The enumeration of the nonlinear terms of f is done similarly.
2. *Period-Testing Algorithm*: which tests whether the generated function f has an optimal period of $2^n - 1$ or not. This algorithm selects a predefined seed (X_0) and starts searching from this point. We prove in Sect. 4.2 that the seed must appear again in the register since it generates a cycle of periodic states as shown in Fig. 3a. The length of the generated cycle is the period. We argue that the seed cannot be outside the state cycle (Fig. 3b). For a given value of n, if the seed X_0 appears after $2^n - 1$ shift operations, then the maximum period of the given function f is achieved, and f is reported as a feedback function with an optimal period. Otherwise, the function f does not have an optimal period.

4.2 Proof of Correctness

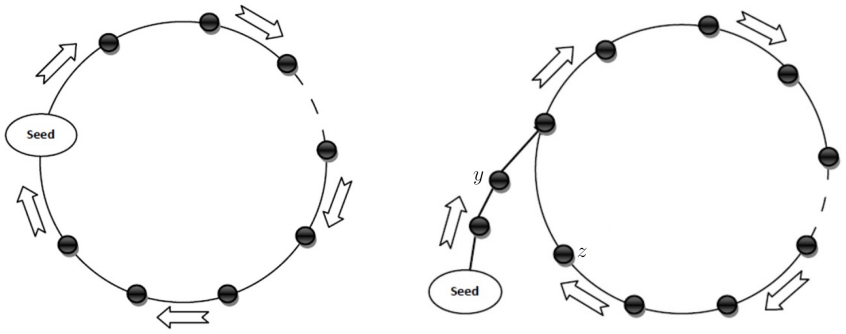
For a seed generating a periodic bit sequence of length p, the seed must also generates p periodic states in the register where each bit in the sequence is the right-most bit in the corresponding state. We examine two scenarios: (A) the seed generates $p-1$ states then it comes back to itself to make a cycle of length p that includes the seed, and (B) the seed generates a number of states outside the cycle, then it generates p states in a cycle that excludes the seed. Figure 3 illustrates these two scenarios.

In order to properly check for NLFSR periods, our Period-Testing algorithm is based on the following observation: Given a feedback function f of some NLFSR, and its size n, it is sufficient to monitor the state of the register to find the period. This observation follows from the following easy-to-prove theorem, which rules out Scenario B.

Theorem 1. *An NLFSR repeats the same bit sequence if and only if the seed appears again in the register.*

Proof. First, if the seed appears again in the register, then proving that the NLFSR will repeat the sequence $s = (s_0, s_1, \dots)$ is straightforward. Second, to prove the converse, let $y = (y_0, y_1, \dots, y_{n-1})$ and $z = (z_0, z_1, \dots, z_{n-1})$ be

two states such that the NLFSR repeats the same sequence $s = (s_0, s_1, \dots)$ if y or z appears in the register (Fig. 3b). Since the first output bit s_0 is the one in the cell 0 of the register, it should be the first bit in y and z, which implies $s_0 = y_0 = z_0$. Similarly, the second output bit $s_1 = y_1 = z_1$ after a shift operation, and hence, the i^{th} output bit $s_{i-1} = y_{i-1} = z_{i-1}$, for $i = 1, 2, \dots, n$. Therefore, the two states y and z generate the same output sequence of bits if and only if $y = z$. □



(a) The seed belongs to the cycle. (b) The seed is outside the cycle.

Fig. 3: Two scenarios for a seed generating periodic states

By construction, the feedback functions generated by the proposed SFG are pairwise nonequivalent. Therefore, they are expected to output nonequivalent pseudorandom bit sequences.

5 New NLFSR Functions with Optimal Periods

In this work, we verify all feedback functions listed by Dubrova [21] for different values of $n \leq 25$ using our Period-Testing algorithm introduced in Sect. 4. We found that all functions give optimal periods. However, the list in [21] is incomplete, and many feedback functions are missing. We extend Dubrova's work by constructing all missing feedback functions with optimal periods for all $n = 4, 5, ..., 19$. These function are pairwise nonequivalent and different from the existing ones.

Table 1 summarizes the improvement achieved by our construction method in each type of the feedback functions considered by Dubrova [21]. The complete lists of all proposed NLFSR feedback functions with optimal periods are given in Sections 5.1, 5.2 and 5.3 for the functions of types 1, 2 and 3 respectively. If a constructed function with optimal period is equivalent to an existing function in [21], it will not be listed here.

Table 1: Number of NLFSR feedback functions for $4 \leq n \leq 19$

Type	Proposed	Existing	Total
1	59	61	120
2	209	209	418
3	157	155	312

5.1 Proposed Functions of Type 1

Our construction method found 59 new functions with optimal periods for $n = 4 \ldots 16$. However, no new functions were found for $n = 17$, 18 or 19. The new feedback functions of type 1 that have optimal periods are listed below. Each entry below is of the form n $(0, a, b, c, d)$ where n is the size of the feedback function, and a, b, c and d are the indices of the variables in type 1 functions defined in Sect. 3.

4 $(0, 2, 3, 1, 2)$
4 $(0, 2, 3, 1, 3)$
4 $(0, 2, 3, 2, 3)$
5 $(0, 2, 4, 1, 4)$
5 $(0, 2, 4, 2, 3)$
5 $(0, 2, 4, 2, 4)$
5 $(0, 3, 4, 1, 3)$
6 $(0, 2, 5, 2, 5)$
6 $(0, 3, 4, 1, 5)$
6 $(0, 3, 4, 2, 4)$
6 $(0, 3, 4, 3, 4)$
6 $(0, 3, 4, 3, 5)$
6 $(0, 3, 5, 1, 5)$
6 $(0, 4, 5, 2, 4)$
6 $(0, 4, 5, 4, 5)$
7 $(0, 2, 6, 1, 3)$
7 $(0, 2, 6, 2, 4)$
7 $(0, 2, 6, 2, 6)$
7 $(0, 3, 5, 2, 5)$
7 $(0, 3, 5, 5, 6)$
7 $(0, 3, 6, 4, 6)$
7 $(0, 5, 6, 1, 5)$
8 $(0, 2, 7, 1, 7)$
8 $(0, 2, 7, 2, 3)$
8 $(0, 2, 7, 3, 4)$
8 $(0, 2, 7, 4, 6)$
8 $(0, 2, 7, 6, 7)$
8 $(0, 3, 6, 1, 5)$
8 $(0, 3, 6, 3, 4)$
8 $(0, 3, 6, 4, 6)$
8 $(0, 3, 7, 3, 7)$
8 $(0, 4, 5, 1, 2)$
8 $(0, 4, 5, 1, 4)$
8 $(0, 4, 5, 1, 6)$
8 $(0, 4, 5, 2, 4)$
8 $(0, 4, 5, 4, 5)$
8 $(0, 4, 5, 4, 6)$
9 $(0, 3, 8, 1, 5)$
9 $(0, 3, 8, 3, 5)$
9 $(0, 5, 6, 2, 6)$
9 $(0, 5, 7, 4, 5)$
10 $(0, 2, 9, 3, 4)$
10 $(0, 5, 6, 1, 6)$
10 $(0, 5, 6, 2, 6)$
10 $(0, 5, 6, 4, 8)$
10 $(0, 5, 8, 5, 9)$
10 $(0, 6, 9, 3, 7)$
10 $(0, 8, 9, 1, 2)$
11 $(0, 2, 10, 7, 10)$
11 $(0, 3, 9, 2, 5)$
11 $(0, 6, 9, 2, 10)$
12 $(0, 4, 9, 3, 9)$
12 $(0, 5, 8, 5, 8)$
12 $(0, 5, 8, 5, 11)$
13 $(0, 2, 12, 4, 8)$
13 $(0, 5, 9, 3, 4)$
14 $(0, 12, 13, 2, 7)$
15 $(0, 6, 10, 4, 13)$
16 $(0, 3, 14, 13, 14)$

5.2 Proposed Functions of Type 2

For type 2 functions, we propose 209 new feedback functions with optimal periods for $n = 4 \ldots 19$. The proposed feedback functions of type 2 are listed below. The entries are of the form n $(0, a, b, c, d, e)$ where n is the size of the function, and a, b, c, d and e are the indices in type 2 functions.

4 $(0,2,1,3,2,3)$
4 $(0,3,1,2,2,3)$
5 $(0,3,1,2,3,4)$
5 $(0,3,1,3,1,4)$
5 $(0,3,1,3,2,4)$
5 $(0,3,1,4,2,3)$
5 $(0,4,1,2,3,4)$
5 $(0,4,1,3,2,4)$
5 $(0,4,1,4,2,3)$
5 $(0,4,2,3,2,4)$
5 $(0,4,2,3,3,4)$
6 $(0,3,2,3,2,5)$
6 $(0,3,2,4,2,5)$
6 $(0,3,2,5,3,4)$
6 $(0,4,1,2,1,5)$
6 $(0,4,1,3,2,3)$
6 $(0,4,1,3,3,4)$
6 $(0,4,1,3,3,5)$
6 $(0,4,1,5,2,4)$
6 $(0,4,2,3,3,5)$
6 $(0,4,2,4,3,5)$
6 $(0,5,1,2,4,5)$
6 $(0,5,1,3,3,5)$
6 $(0,5,1,4,3,4)$
7 $(0,4,1,4,3,6)$
7 $(0,4,1,6,3,4)$
7 $(0,4,1,6,4,6)$
7 $(0,4,2,3,4,5)$
7 $(0,4,2,4,2,5)$
7 $(0,4,2,4,2,6)$
7 $(0,4,4,5,5,6)$
7 $(0,5,1,2,1,4)$
7 $(0,5,1,2,1,6)$
7 $(0,5,1,3,1,5)$
7 $(0,5,1,3,2,5)$
7 $(0,5,1,3,5,6)$
7 $(0,5,1,4,1,6)$
7 $(0,5,1,5,2,6)$
7 $(0,5,1,6,3,5)$
7 $(0,5,2,4,3,5)$
7 $(0,5,3,4,3,6)$
7 $(0,6,1,2,5,6)$
7 $(0,6,1,6,2,3)$
7 $(0,6,2,3,2,5)$
7 $(0,6,2,3,3,4)$
7 $(0,6,2,4,2,5)$
7 $(0,6,2,4,4,5)$
7 $(0,6,2,6,3,4)$
8 $(0,5,1,2,4,6)$
8 $(0,5,1,4,2,7)$
8 $(0,5,1,5,2,6)$
8 $(0,5,2,3,5,6)$
8 $(0,5,2,4,2,7)$
8 $(0,5,2,5,2,7)$
8 $(0,5,4,6,4,7)$
8 $(0,5,4,6,6,7)$
8 $(0,6,1,2,1,5)$
8 $(0,6,1,2,3,7)$
8 $(0,6,1,3,5,7)$
8 $(0,6,1,7,5,6)$
8 $(0,6,2,4,5,7)$
8 $(0,7,1,2,4,6)$
8 $(0,7,1,4,4,5)$
8 $(0,7,2,7,3,6)$
8 $(0,7,4,6,4,7)$
8 $(0,7,4,6,5,6)$
9 $(0,5,1,4,6,7)$
9 $(0,5,1,6,2,7)$
9 $(0,5,1,7,2,3)$
9 $(0,5,1,7,4,7)$
9 $(0,5,1,7,6,8)$
9 $(0,5,2,6,4,6)$
9 $(0,5,3,6,3,8)$
9 $(0,6,1,5,2,8)$
9 $(0,6,2,5,6,7)$
9 $(0,6,2,5,7,8)$
9 $(0,6,2,8,3,8)$
9 $(0,7,1,8,5,6)$
9 $(0,7,2,5,3,4)$
9 $(0,7,2,7,3,5)$
9 $(0,7,2,7,3,8)$
9 $(0,7,3,5,4,8)$
9 $(0,8,1,4,2,6)$
9 $(0,8,1,6,6,7)$
9 $(0,8,1,7,2,6)$
9 $(0,8,1,8,2,7)$
9 $(0,8,1,8,3,4)$
9 $(0,8,2,3,3,8)$
9 $(0,8,4,6,5,6)$
9 $(0,8,4,7,4,8)$
10 $(0,6,1,3,2,3)$
10 $(0,6,1,3,5,9)$
10 $(0,6,1,3,7,9)$
10 $(0,6,1,9,5,9)$
10 $(0,6,2,3,7,9)$
10 $(0,6,3,9,7,9)$
10 $(0,7,1,2,3,4)$
10 $(0,7,1,2,3,8)$
10 $(0,7,1,3,2,8)$
10 $(0,7,1,3,7,9)$
10 $(0,7,1,4,4,9)$
10 $(0,7,2,7,4,9)$
10 $(0,7,2,8,8,9)$
10 $(0,7,4,8,7,8)$
10 $(0,8,1,2,1,7)$
10 $(0,8,1,5,1,9)$
10 $(0,8,1,6,5,9)$
10 $(0,8,3,6,7,9)$
10 $(0,8,3,7,5,7)$
10 $(0,8,3,7,6,9)$
10 $(0,8,3,9,4,6)$
10 $(0,8,3,9,4,9)$
10 $(0,8,5,7,5,9)$
10 $(0,9,1,3,2,8)$
10 $(0,9,2,6,3,4)$
10 $(0,9,2,7,3,6)$
10 $(0,9,5,8,6,8)$
10 $(0,9,6,7,8,9)$
11 $(0,6,2,5,7,10)$
11 $(0,6,3,5,3,9)$
11 $(0,6,4,5,4,7)$
11 $(0,7,1,2,9,10)$
11 $(0,7,1,9,8,9)$
11 $(0,7,3,7,4,8)$
11 $(0,8,1,4,4,8)$
11 $(0,8,1,6,2,10)$
11 $(0,8,2,3,5,10)$
11 $(0,8,2,5,6,8)$
11 $(0,8,3,6,5,8)$
11 $(0,8,4,6,4,9)$
11 $(0,9,1,2,5,7)$
11 $(0,9,1,3,2,4)$
11 $(0,9,1,5,6,7)$
11 $(0,10,1,8,4,9)$
11 $(0,10,2,10,4,9)$

11 $(0, 10, 3, 4, 4, 8)$
11 $(0, 10, 3, 8, 4, 8)$
11 $(0, 10, 3, 9, 4, 10)$
11 $(0, 10, 6, 7, 8, 9)$
11 $(0, 10, 6, 9, 7, 8)$
12 $(0, 9, 1, 7, 2, 6)$
12 $(0, 9, 2, 5, 4, 11)$
12 $(0, 10, 1, 4, 1, 7)$
12 $(0, 10, 1, 5, 3, 5)$
12 $(0, 10, 1, 9, 3, 9)$
12 $(0, 10, 1, 11, 2, 11)$
12 $(0, 10, 3, 5, 9, 10)$
12 $(0, 10, 3, 7, 3, 9)$
12 $(0, 10, 4, 10, 5, 11)$
12 $(0, 10, 6, 9, 9, 11)$
12 $(0, 11, 1, 4, 2, 3)$
12 $(0, 11, 2, 4, 4, 5)$
12 $(0, 11, 2, 6, 4, 10)$
12 $(0, 11, 2, 9, 7, 10)$
12 $(0, 11, 8, 9, 9, 10)$
13 $(0, 7, 1, 11, 8, 12)$
13 $(0, 8, 1, 10, 2, 4)$
13 $(0, 8, 1, 12, 2, 6)$
13 $(0, 8, 4, 9, 7, 10)$
13 $(0, 8, 4, 9, 8, 12)$
13 $(0, 8, 4, 11, 8, 9)$
13 $(0, 9, 1, 3, 10, 12)$
13 $(0, 9, 3, 5, 4, 11)$
13 $(0, 9, 7, 9, 10, 12)$
13 $(0, 10, 2, 4, 7, 9)$
13 $(0, 10, 2, 11, 4, 12)$
13 $(0, 10, 3, 4, 4, 5)$
13 $(0, 11, 1, 3, 3, 6)$
13 $(0, 11, 1, 4, 7, 12)$
13 $(0, 12, 2, 7, 10, 11)$
13 $(0, 12, 2, 8, 8, 11)$
13 $(0, 12, 4, 11, 8, 9)$
13 $(0, 12, 5, 7, 7, 11)$
13 $(0, 12, 6, 10, 6, 12)$
14 $(0, 8, 1, 2, 5, 9)$
14 $(0, 8, 1, 13, 5, 9)$
14 $(0, 9, 1, 8, 10, 12)$
14 $(0, 11, 1, 8, 2, 12)$
14 $(0, 11, 2, 7, 4, 9)$
14 $(0, 11, 2, 8, 10, 12)$
14 $(0, 11, 2, 10, 8, 13)$
14 $(0, 12, 3, 11, 9, 13)$
14 $(0, 13, 1, 12, 2, 10)$
14 $(0, 13, 2, 5, 2, 9)$
15 $(0, 8, 2, 5, 4, 13)$
15 $(0, 8, 2, 12, 3, 12)$
15 $(0, 9, 2, 3, 4, 7)$
15 $(0, 11, 1, 3, 7, 8)$
15 $(0, 11, 1, 10, 9, 10)$
15 $(0, 11, 4, 13, 5, 8)$
15 $(0, 11, 5, 6, 5, 9)$
15 $(0, 13, 1, 3, 7, 9)$
16 $(0, 9, 2, 8, 4, 5)$
16 $(0, 9, 3, 6, 10, 14)$
16 $(0, 11, 4, 10, 8, 12)$
16 $(0, 11, 4, 12, 8, 9)$
16 $(0, 13, 2, 9, 3, 14)$
16 $(0, 13, 9, 11, 11, 15)$
17 $(0, 9, 1, 6, 5, 7)$
17 $(0, 10, 3, 8, 9, 16)$
17 $(0, 11, 5, 10, 8, 15)$
17 $(0, 12, 4, 11, 10, 13)$
17 $(0, 14, 3, 4, 8, 11)$
17 $(0, 16, 2, 8, 7, 10)$
18 $(0, 15, 7, 16, 11, 12)$
18 $(0, 17, 6, 7, 6, 9)$
19 $(0, 13, 1, 2, 11, 15)$
19 $(0, 15, 3, 14, 5, 12)$
19 $(0, 17, 9, 11, 11, 13)$

5.3 Proposed Functions of Type 3

We propose 157 new functions of type 3 with optimal periods for $n = 6 \ldots 19$. There are no new functions for $n = 4$ or 5. The new feedback functions are listed below. Each entry is of the form n $(0, a, b, c, d, e, h)$ where n is the size of the function, and a, b, c, d, e and h are the indices in type 3 functions.

6 $(0, 1, 2, 3, 4, 1, 5)$
6 $(0, 1, 2, 3, 4, 3, 5)$
6 $(0, 1, 2, 3, 5, 1, 4)$
6 $(0, 1, 2, 3, 5, 2, 3)$
6 $(0, 1, 3, 4, 5, 2, 5)$
6 $(0, 1, 3, 4, 5, 3, 4)$
6 $(0, 2, 3, 4, 5, 1, 3)$
6 $(0, 2, 3, 4, 5, 1, 5)$
7 $(0, 1, 4, 5, 6, 1, 5)$
7 $(0, 1, 4, 5, 6, 2, 6)$
7 $(0, 1, 4, 5, 6, 4, 6)$
7 $(0, 2, 3, 5, 6, 1, 5)$
7 $(0, 2, 3, 5, 6, 2, 6)$
7 $(0, 2, 3, 5, 6, 5, 6)$
7 $(0, 3, 4, 5, 6, 1, 5)$
7 $(0, 3, 4, 5, 6, 1, 6)$
7 $(0, 3, 4, 5, 6, 4, 5)$
8 $(0, 1, 3, 6, 7, 4, 6)$
8 $(0, 1, 4, 5, 7, 1, 5)$
8 $(0, 1, 4, 5, 7, 2, 7)$
8 $(0, 1, 4, 5, 7, 4, 7)$
8 $(0, 1, 4, 6, 7, 3, 7)$
8 $(0, 2, 3, 4, 5, 2, 6)$
8 $(0, 2, 3, 4, 6, 4, 5)$
8 $(0, 2, 4, 5, 6, 3, 4)$
8 $(0, 2, 5, 6, 7, 1, 3)$
8 $(0, 2, 5, 6, 7, 3, 5)$
8 $(0, 3, 4, 5, 6, 2, 6)$
8 $(0, 3, 4, 6, 7, 4, 6)$
8 $(0, 3, 5, 6, 7, 2, 6)$
9 $(0, 1, 4, 6, 8, 4, 6)$
9 $(0, 1, 4, 7, 8, 3, 7)$
9 $(0, 2, 3, 5, 8, 2, 8)$
9 $(0, 2, 3, 7, 8, 3, 6)$
9 $(0, 2, 4, 6, 8, 3, 4)$
9 $(0, 2, 5, 6, 7, 1, 7)$

9 (0, 2, 5, 7, 8, 3, 8)
9 (0, 2, 6, 7, 8, 3, 5)
9 (0, 3, 4, 7, 8, 3, 7)
9 (0, 3, 4, 7, 8, 3, 8)
9 (0, 4, 5, 6, 8, 2, 6)
9 (0, 5, 6, 7, 8, 2, 6)
10 (0, 1, 5, 6, 9, 1, 5)
10 (0, 1, 5, 6, 9, 1, 6)
10 (0, 1, 5, 6, 9, 1, 9)
10 (0, 2, 5, 6, 8, 6, 8)
10 (0, 2, 5, 8, 9, 1, 5)
10 (0, 2, 6, 8, 9, 5, 9)
10 (0, 2, 6, 8, 9, 6, 8)
10 (0, 3, 4, 5, 6, 2, 6)
10 (0, 3, 4, 5, 7, 2, 6)
10 (0, 3, 4, 5, 7, 4, 6)
10 (0, 3, 4, 5, 9, 2, 8)
10 (0, 3, 4, 6, 8, 4, 9)
10 (0, 3, 4, 7, 9, 4, 9)
10 (0, 3, 5, 6, 7, 4, 6)
10 (0, 3, 5, 6, 7, 4, 8)
10 (0, 3, 6, 7, 9, 4, 7)
10 (0, 4, 5, 6, 7, 4, 8)
10 (0, 4, 6, 7, 8, 4, 7)
11 (0, 1, 4, 7, 10, 2, 10)
11 (0, 1, 5, 8, 10, 2, 4)
11 (0, 1, 7, 8, 10, 1, 5)
11 (0, 1, 7, 9, 10, 2, 8)
11 (0, 1, 7, 9, 10, 2, 10)
11 (0, 2, 3, 5, 10, 5, 9)
11 (0, 2, 4, 6, 10, 3, 9)
11 (0, 2, 4, 8, 10, 3, 10)
11 (0, 2, 6, 7, 9, 2, 6)
11 (0, 3, 4, 8, 9, 1, 4)
11 (0, 3, 4, 8, 9, 1, 5)
11 (0, 3, 4, 8, 9, 1, 7)
11 (0, 3, 4, 9, 10, 1, 2)
11 (0, 3, 4, 9, 10, 2, 10)
11 (0, 3, 5, 6, 10, 2, 6)
11 (0, 3, 5, 8, 10, 3, 5)
11 (0, 3, 6, 7, 10, 4, 6)
11 (0, 4, 5, 6, 8, 3, 7)
11 (0, 4, 5, 7, 8, 8, 9)
11 (0, 4, 7, 9, 10, 2, 3)
11 (0, 4, 7, 9, 10, 2, 7)
11 (0, 4, 7, 9, 10, 8, 9)
11 (0, 5, 6, 7, 8, 1, 9)
11 (0, 6, 7, 9, 10, 5, 7)
11 (0, 6, 8, 9, 10, 5, 7)
12 (0, 1, 6, 7, 11, 4, 5)
12 (0, 1, 6, 10, 11, 6, 10)
12 (0, 2, 3, 7, 11, 5, 6)
12 (0, 2, 4, 8, 11, 7, 10)
12 (0, 2, 5, 9, 11, 7, 8)
12 (0, 2, 6, 7, 10, 2, 10)
12 (0, 2, 6, 9, 10, 2, 8)
12 (0, 2, 6, 9, 10, 6, 10)
12 (0, 2, 8, 9, 10, 4, 9)
12 (0, 3, 5, 7, 11, 1, 11)
12 (0, 3, 6, 9, 11, 2, 8)
12 (0, 3, 6, 9, 11, 3, 11)
12 (0, 3, 7, 10, 11, 1, 5)
12 (0, 4, 6, 7, 11, 2, 6)
12 (0, 4, 6, 7, 11, 6, 8)
12 (0, 4, 6, 9, 10, 6, 9)
12 (0, 5, 6, 9, 11, 2, 8)
12 (0, 7, 9, 10, 11, 3, 7)
13 (0, 2, 3, 6, 12, 7, 11)
13 (0, 2, 3, 11, 12, 1, 7)
13 (0, 2, 5, 10, 11, 3, 12)
13 (0, 2, 6, 8, 12, 3, 5)
13 (0, 2, 7, 8, 11, 2, 5)
13 (0, 3, 6, 7, 11, 1, 5)
13 (0, 3, 6, 8, 10, 3, 11)
13 (0, 3, 7, 8, 10, 2, 5)
13 (0, 3, 8, 9, 12, 5, 9)
13 (0, 4, 6, 8, 12, 4, 5)
13 (0, 6, 7, 8, 12, 4, 8)
13 (0, 7, 9, 10, 12, 3, 7)
13 (0, 8, 9, 11, 12, 6, 12)
14 (0, 1, 9, 10, 13, 8, 13)
14 (0, 2, 4, 8, 13, 5, 7)
14 (0, 2, 4, 8, 13, 7, 11)
14 (0, 2, 4, 11, 12, 4, 5)
14 (0, 2, 5, 7, 13, 1, 11)
14 (0, 2, 8, 9, 12, 4, 8)
14 (0, 3, 5, 7, 12, 2, 3)
14 (0, 3, 7, 10, 13, 3, 13)
14 (0, 4, 7, 8, 10, 1, 9)
14 (0, 6, 8, 9, 10, 10, 13)
14 (0, 6, 9, 10, 13, 6, 12)
14 (0, 7, 9, 11, 12, 9, 13)
14 (0, 7, 10, 12, 13, 11, 13)
14 (0, 9, 11, 12, 13, 11, 13)
15 (0, 2, 6, 12, 13, 8, 12)
15 (0, 3, 10, 11, 14, 11, 12)
15 (0, 5, 8, 10, 11, 1, 14)
15 (0, 5, 10, 11, 12, 8, 12)
15 (0, 7, 8, 10, 12, 2, 12)
16 (0, 2, 3, 11, 15, 1, 2)
16 (0, 2, 6, 11, 14, 2, 10)
16 (0, 3, 4, 5, 15, 1, 11)
16 (0, 3, 8, 9, 14, 1, 13)
16 (0, 4, 5, 10, 14, 1, 2)
16 (0, 6, 7, 8, 12, 4, 8)
16 (0, 6, 8, 9, 14, 10, 13)
16 (0, 7, 13, 14, 15, 2, 10)
17 (0, 3, 5, 14, 16, 7, 15)
17 (0, 5, 8, 13, 15, 1, 11)
17 (0, 5, 8, 14, 16, 4, 10)
17 (0, 6, 8, 12, 16, 4, 16)
17 (0, 7, 10, 11, 14, 2, 8)
18 (0, 3, 6, 12, 16, 3, 7)
18 (0, 5, 7, 9, 15, 2, 14)
18 (0, 6, 7, 13, 17, 9, 17)
18 (0, 7, 9, 10, 17, 3, 16)
18 (0, 11, 13, 14, 17, 8, 16)
19 (0, 2, 11, 15, 18, 6, 18)
19 (0, 3, 10, 12, 16, 2, 16)
19 (0, 5, 7, 13, 14, 1, 17)
19 (0, 11, 14, 15, 18, 4, 14)

6 Conclusion

In this work, we verified previous study done by Dubrova which provides a list of NLFSR feedback functions in the range ($4 \leq n < 25$). These functions are of three predefined types, and they all have the optimal period of $2^n - 1$ where n is the size of the register. Since Dubrova's work did not list all feedback functions for many values of n in the specified range, we closed the gap by proposing the missing feedback functions with optimal periods for all $n = 4, 5, \cdots, 19$. The construction method of these functions is briefly explained. As for future work, we suggest extending this work in two ways: (1) construct feedback functions with larger values of n, and (2) explore different types of feedback functions and search for new types with more functions with optimal periods. Since all feedback functions considered in this work are of degree 2, we may also extend the search to include degree 3 functions.

Acknowledgment

The authors would like to thank King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, for supporting this research. Figures and descriptions in this paper were provided by the authors and are used with permission.

References

1. G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generators-new devices for embedded test applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 9, pp. 1306–1320, 2004.
2. K. Zeng, C.-H. Yang, D.-Y. Wei, and T. Rao, "Pseudorandom bit generators in stream-cipher cryptography," *Computer*, vol. 24, no. 2, pp. 8–17, 1991.
3. J. McCluskey, "High speed calculation of cyclic redundancy codes," in *Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays*. ACM, 1999, p. 250.
4. A. Ahmad, "Achievement of higher testability goals through the modification of shift registers in LFSR-based testing," *International journal of electronics*, vol. 82, no. 3, pp. 249–260, 1997.
5. S. Golomb, "Shift register sequences. laguna hills, ca aegean," 1982.
6. C. J. A. Jansen, "Investigations on nonlinear streamcipher systems: construction and evaluation methods," 1989.
7. E. Dubrova, "Generation of full cycles by a composition of nlfsrs," *Designs, codes and cryptography*, vol. 73, no. 2, pp. 469–486, 2014.
8. T. Rachwalik, J. Szmidt, R. Wicik, and J. Zabłocki, "Generation of nonlinear feedback shift registers with special-purpose hardware," in *Communications and Information Systems Conference (MCC), 2012 Military*. IEEE, 2012, pp. 1–4.
9. C. F. Sainte-Marie, "Solution to question nr. 48," *LâĂŹintermédiaire des Mathématiciens*, vol. 1, pp. 107–110, 1894.
10. F. de Bruijn, "A combinatorial problem," 1946.
11. G. L. Mayhew and S. W. Golomb, "Linear spans of modified de bruijn sequences," *IEEE transactions on information theory*, vol. 36, no. 5, pp. 1166–1167, 1990.

12. B. M. Gammel, R. Göttfert, and O. Kniffler, "The achterbahn stream cipher," *Submission to eSTREAM*, 2005.
13. R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge university press, 1994.
14. J. Massey, "Shift-register synthesis and BCH decoding," *IEEE transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.
15. E. Dubrova, "A transformation from the fibonacci to the galois nlfsrs," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5263–5271, 2009.
16. J.-M. Chabloz, S. S. Mansouri, and E. Dubrova, "An algorithm for constructing a fastest galois nlfsr generating a given sequence." in *SETA*. Springer, 2010, pp. 41–54.
17. E. Dubrova, "Finding matching initial states for equivalent nlfsrs in the fibonacci and the galois configurations," *IEEE transactions on information theory*, vol. 56, no. 6, pp. 2961–2966, 2010.
18. M. Liu, S. S. Mansouri, and E. Dubrova, "A faster shift register alternative to filter generators," in *Digital System Design (DSD), 2013 Euromicro Conference on*. IEEE, 2013, pp. 713–718.
19. A. Castro Lechtaler, M. Cipriano, E. García, J. Liporace, A. Maiorano, and E. Malvacio, "Model design for a reduced variant of a trivium type stream cipher," *Journal of Computer Science & Technology*, vol. 14, 2014.
20. H. Fredricksen, "A survey of full length nonlinear shift register cycle algorithms," *SIAM review*, vol. 24, no. 2, pp. 195–221, 1982.
21. E. Dubrova, "A list of maximum-period nlfsrs," 2012.
22. K. Mandal and G. Gong, "Probabilistic generation of good span n sequences from nonlinear feedback shift registers," *University of Waterloo*, 2012.
23. I. J. Good, "Normal recurring decimals," *Journal of the London Mathematical Society*, vol. 1, no. 3, pp. 167–169, 1946.
24. R. Motwani and P. Raghavan, "Randomized algorithms. cambridge international series on parallel computation," 1995.
25. C. Banderier and R. P. Dobrow, "A generalized cover time for random walks on graphs," in *Formal Power Series and Algebraic Combinatorics*. Springer, 2000, pp. 113–124.