

**NEW NON-LINEAR FEEDBACK SHIFT REGISTER
FUNCTIONS WITH OPTIMAL PERIODS**

BY
IBRAHEEM MOHAMMED AL-HEJRI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In
INFORMATION ASSURANCE & SECURITY

APRIL 2018

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

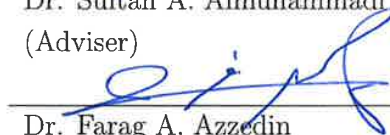
DEANSHIP OF GRADUATE STUDIES

This thesis, written by **IBRAHEEM MOHAMMED AL-HEJRI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN INFORMATION ASSURANCE & SECURITY**.

Thesis Committee



Dr. Sultan A. Almuhammadi
(Adviser)



Dr. Farag A. Azzedin
(Member)



Dr. Muhammad Y. Mahmoud
(Member)



Dr. Khalid A. Aljasser
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

Date

14/5/18



©Ibraheem Mohammed Al-Hejri
2018

I dedicate my thesis work to my family and friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. I also wish to dedicate this work to brothers and friends. This work would not be possible without their support. I would also like to sincerely thank all teachers who taught me, as their encouragements and motivations have laid the foundation for this work.

ACKNOWLEDGMENTS

At first, I am grateful to the almighty Allah for giving me ability to accomplish this thesis. Without His mercy and blessing, nothing is possible. I wish to give my sincere appreciation to my advisor Dr. Sultan Almuhammadi for all his efforts and countless hours of reflecting, reading, encouraging, guiding and most of all patience throughout the entire course of this thesis. He has been my mentor, teacher and father. I feel very lucky to have him as a supervisor for my thesis. I would also like to thank my thesis committee members, Dr.Farag Azzedin and Dr.Muhammad Mahmoud for their valuable cooperation and constructive comments throughout my thesis. They have dedicated valuable time out of their busy schedules in order to provide useful feedback to improve the quality of this work. I acknowledge the Information Technology Center at King Fahd University of Petroleum and Minerals for providing high performance computing resources that have contributed to the research results reported within this work. Last but not least, I am immensely indebted to my parents and brothers for their endless love and moral support.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	iii
LIST OF TABLES	vi
LIST OF FIGURES	viii
ABSTRACT (ENGLISH)	ix
ABSTRACT (ARABIC)	xi
CHAPTER 1 INTRODUCTION	1
1.1 Research Objectives	3
1.2 Research Methodology	4
1.3 Motivations	5
1.4 Contributions	6
1.5 Thesis Outline	7
CHAPTER 2 BACKGROUND	8
2.1 Feedback Shift Registers (<i>FSR</i>)	8
2.1.1 Linear Feedback Shift Register (<i>LFSR</i>)	9
2.1.2 Non- Linear Feedback Shift Register (<i>NLFSR</i>)	11
2.2 Applications of <i>LFSR</i> and <i>NLFSR</i>	13
2.3 Security of <i>NLFSR</i>	15
CHAPTER 3 RELATED WORK	19

3.1	Exploring <i>NLFSR</i> Feedback Functions	20
3.2	Using <i>NLFSR</i> Feedback Functions	23
3.3	Summary	29
CHAPTER 4 METHODOLOGY		32
4.1	Preliminaries	32
4.2	Environmental Setup	35
4.3	Proposed Construction Method	36
4.4	Proof of Correctness	40
CHAPTER 5 PROPOSED <i>NLFSR</i> FUNCTIONS WITH OPTI-		
MAL PERIODS		42
5.1	Constructing New <i>NLFSR</i> Feedback Functions	44
5.1.1	Proposed Functions of Type 1	44
5.1.2	Proposed Functions of Type 2	47
5.1.3	Proposed Functions of Type 3	50
5.1.4	Proposed Functions of Type 4	52
5.2	Proposing New Type of <i>NLFSR</i> Feedback Functions	55
5.2.1	Proposed New Feedback Functions of Type 5	56
5.2.2	List of the Proposed Feedback Functions of Type 5	60
CHAPTER 6 CONCLUSION AND FUTURE WORK		86
6.1	Thesis Summary	87
6.2	Concluding Remarks	89
6.3	Future Work	90
CHAPTER 7 APPENDIX		91
7.1	List of Proposed <i>NLFSR</i> Feedback Functions with Optimal Periods	92
REFERENCES		101
VITAE		108

LIST OF TABLES

2.1	A Comparison Between <i>LFSR</i> and <i>NLFSR</i>	18
3.1	Number of <i>NLFSR</i> Feedback Functions for $(4 \leq n \leq 24)$ Proposed by Dubrova [1]	22
3.2	The Strengths and the Weaknesses of the Studies Focused on Ex- ploring <i>NLFSR</i> Feedback Functions	22
3.3	The Strengths and the Weaknesses of the Studies Using <i>NLFSR</i> Feedback Functions	29
3.4	Summary of Existing Studies on <i>NLFSR</i> Functions with Optimal Periods	31
4.1	Experimental Setup	35
5.1	Number of <i>NLFSR</i> Feedback Functions for $(4 \leq n \leq 25)$	43
5.2	Summary of the New Feedback Functions of <i>NLFSRs</i> with Optimal Periods for $(4 \leq n \leq 25)$	43
5.3	Examples of Type 1 Functions with Optimal Periods	45
5.4	Examples of Type 2 Functions with Optimal Periods	48
5.5	Examples of Type 3 Functions with Optimal Periods	50
5.6	Examples of Type 4 Functions with Optimal Periods	53
5.7	Examples of Type 5 Functions with Optimal Periods	56
5.8	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 5$	60
5.9	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 6$	61
5.10	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 7$	62

5.11	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 8$	64
5.12	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 9$	66
5.13	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 10$	68
5.14	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 11$	70
5.15	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 12$	72
5.16	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 13$	74
5.17	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 14$	76
5.18	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 15$	78
5.19	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 16$	79
5.20	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 17$	80
5.21	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 18$	81
5.22	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 19$	82
5.23	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 20$	83
5.24	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 21$	83
5.25	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 22$	84
5.26	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 23$	85
5.27	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 24$	85
5.28	<i>NLFSR</i> Functions (Type 5) with Optimal Periods, $n = 25$	85
7.1	<i>NLFSR</i> Functions with Optimal Periods for $(4 \leq n \leq 25)$	92

LIST OF FIGURES

2.1	The General Structure of an n -bit $LFSR$ [2].	10
2.2	The General Structure of an n -bit $NLFSR$ [1].	11
2.3	A 4-bit Fibonacci $NLFSR$ Example [1].	12
3.1	An Approach for Generating Keystream Using $NLFSRs$ [3]. . . .	24
4.1	The Seed Belongs to the Cycle.	37
4.2	The Seed is Outside the Cycle.	38
5.1	An Example of Feedback Function of Type 1	46
5.2	Total Number of Feedback Functions (Proposed and Existing) of Type 1.	47
5.3	Total Number of Feedback Functions (Proposed and Existing) of Type 2.	49
5.4	Total Number of Feedback Functions (Proposed and Existing) of Type 3.	51
5.5	Total Number of Feedback Functions (Proposed and Existing) of Type 4.	54
5.6	Exploring New $NLFSR$ Feedback Functions of Type 5.	58
5.7	Total Number of Feedback Functions (Proposed and Existing) For all Types.	58
6.1	Taxonomy of Thesis Contributions.	89

THESIS ABSTRACT

NAME: Ibraheem Mohammed Al-Hejri
TITLE OF STUDY: NEW NON-LINEAR FEEDBACK SHIFT REGISTER
FUNCTIONS WITH OPTIMAL PERIODS
MAJOR FIELD: Information Assurance & Security
DATE OF DEGREE: April 2018

Information security is of fundamental importance to many organizations in our community. Cryptographic techniques are used to protect data from unauthorized access. One of the most common components in cryptographic techniques is the use of pseudorandom bit generators. There are several types of pseudorandom bit generators used in different cryptographic algorithms. In recent years, nonlinear feedback shift registers (*NLFSRs*) have received much attention in designing various cryptographic algorithms such as stream ciphers and lightweight block ciphers to provide high-level security in communication systems. In general, *NLFSRs* have much greater linear complexity than linear feedback shift registers (*LFSRs*) of the same period. However, there is no mathematical foundation on

how to construct an *NLFSR* feedback function with optimal period. Therefore, the problem of finding new functions for *NLFSR* of arbitrary size n that give optimal periods remains challenging.

The optimal periods of any feedback shift register grow exponentially in terms of the size of the register, n . Therefore, it is very beneficial to find feedback functions with optimal periods for large n . This work shows the importance of *NLFSRs* and their security strength in recent applications as compared to *LFSRs*. Moreover, it constructs new feedback functions with optimal periods for large values of n , and verifies existing functions of various register sizes, n , that provide optimal periods for *NLFSRs*.

خلاصة الرسالة

الاسم: إبراهيم محمد أحمد المبري
عنوان الرسالة: دوال جديدة ذوات الفترة المثالية لمسجل الإزاحة غير الخطية ذات التغذية الراجعة
التخصص: علوم الحاسب الآلي – أمن و ضمان المعلومات
تاريخ التخرج: شعبان 1439 هـ

يعتبر أمن المعلومات أمراً ذو أهمية أساسية للعديد من المنظمات في مجتمعنا. تُستخدم العديد من تقنيات التشفير من أجل حماية البيانات من الوصول غير المصرح به. يُعتبر استخدام مولدات الشفرة شبه العشوائية إحدى الطرق/ الأدوات الأكثر شيوعاً في تقنيات التشفير. يُوجد هناك عدة أنواع من مولدات الشفرة شبه العشوائية المستخدمة في العديد من خوارزميات التشفير المختلفة. خلال السنوات الأخيرة، حازت سجلات الإزاحة غير الخطية ذات التغذية الراجعة (NLFSRs) على الكثير من الاهتمام في تصميم مختلف خوارزميات التشفير مثل خوارزميات شفرة التدفق (Stream Cipher) أو خوارزميات تشفير الكتلة (Block Cipher) الخفيفة لتوفير الأمن على مستوى عالٍ في أنظمة الاتصالات. بشكلٍ عام، تعتبر سجلات الإزاحة غير الخطية ذات التغذية الراجعة أكثر تعقيداً من سجلات الإزاحة الخطية ذات التغذية الراجعة (LFSRs) لنفس الفترة. ومع ذلك، لا يوجد هناك أساس رياضي لكيفية تركيب/ إنشاء دالة التغذية الراجعة ذات الفترة المثلى لمسجل الإزاحة غير الخطية. ولذلك، فإن مشكلة إيجاد دوال جديدة ذات الحجم العشوائي لسجلات الإزاحة غير الخطية ذات التغذية الراجعة، والتي تُعطي فترات مثالية، لا تزال

تشكل تحدياً صعباً. إن الفترات المثالية لأي مسجل إزاحة غير خطية ذات تغذية راجعة تنمو بإطراد من حيث حجم المسجل. لذلك، من المفيد جداً العثور على دوال تغذية راجعة ذوات فترات مثالية لحجم مسجل كبير. هذا العمل يوضح أهمية سجلات الإزاحة غير الخطية ذات التغذية الراجعة وقوتها الأمنية في التطبيقات الحديثة بالمقارنة مع سجلات الإزاحة الخطية ذات التغذية الراجعة، علاوةً على ذلك، هذا العمل يقوم بتركيب/ إنشاء دوال تغذية راجعة بفترات مثالية لمسجلات ذات حجم كبير، بالإضافة إلى التحقق من الدوال الحالية لمختلف الأحجام، والتي تدعم سجلات الإزاحة غير الخطية ذات التغذية الراجعة بالفترات المثالية.

درجة الماجستير في أمن و ضمان المعلومات
جامعة الملك فهد للبترول والمعادن
الظهران - المملكة العربية السعودية
شعبان 1439 هـ

CHAPTER 1

INTRODUCTION

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. Several security applications require the property of randomness to remain secure. There are many applications of random numbers, like generation of keys for public key cryptography and digital signature algorithms, generation of a symmetric key for use as a temporary key which is used in several network applications such as TLS protocol, Wi-Fi, E-mail security and IP security. Moreover, random numbers can be used in many key distribution protocols such as Kerberos protocol and initialization values (IVs) for handshaking in order to prevent replay attacks and other types of attacks that target the network protocols.

Feedback shift register (*FSR*) is one kind of implementation which has the property of randomness. It can be implemented in either software or hardware form. Depending on its internal feedback function, the *FSR* can be either *linear* (*LFSR*)

or *nonlinear* (*NLFSR*).

An *LFSR* is a shift register that is used to generate pseudorandom sequences. Although *LFSR* has several advantages, including simplicity, speed, the capability of generating a maximal cycle sequence and ease of implementation, it has critical drawbacks. Its linearity and the simplicity of its structure lead to a relatively easy cryptanalysis. The area of *LFSRs* is considered well-known and most of the *LFSR* fundamental problems are solved. A designer only needs to use a primitive generator polynomial in order to build a *LFSR* of size n bits with a maximum period.

An *NLFSR* is a generalization of *LFSR* in which a current state of the register is an output of the nonlinear transformation of the previous state [4]. The importance of *NLFSR* comes from its ability to produce a very secure pseudorandom sequence which is hard to break. To determine the structure of *NLFSR*, at least $\theta(2^n)$ sequence bits are needed, where n is the register size [5], while breaking an *LFSR* just needs a sequence of $\theta(n)$ bits. While the theory behind *LFSR* is well understood, *NLFSRs* still have many fundamental problems, which remain open even with well-understood theory. A systematic method to construct *NLFSRs*, which generates the longest period, is the most significant problem to be solved. This is a challenging problem since there is no mathematical foundation supporting it.

In this work, the importance of *NLFSRs* and their security strength in modern applications are studied and compared to the *LFSRs*. Known *NLFSR* functions with optimal periods are verified. Moreover, new feedback functions that provide optimal periods for *NLFSRs* are constructed for large values of n .

1.1 Research Objectives

The main goals of the present research are the following:

- *Verifying the existing functions of various register sizes, n , that provide optimal periods for NLFSRs:* To achieve this objective, we need to implement an *NLFSR* verifier system to test a list of feedback functions and to find their periods. We need to verify the existing feedback functions of various sizes, n , that provide optimal periods for *NLFSRs* proposed by Dubrova [1]. This also helps verifying the correctness of the implementation of our *NLFSR* period verifying system.
- *Constructing new feedback functions with optimal periods for large values of n , to ensure greater security and immunity against cryptanalytic techniques:* As a result of our verification phase that was done to achieve the first objective of our work, we found that Dubrova's work did not list all the feedback functions having optimal periods for many values of n . Therefore, the goal of this work is to close the gap by constructing the missing feedback functions with optimal periods in the previous studies for all $n = 4, 5, \dots, 25$. Moreover, this work aims to propose a new type of feedback function, that

also provides optimal periods for *NLFSRs* for different values of $n \leq 25$ to ensure these feedback functions are the only ones existing within this range.

1.2 Research Methodology

The goal of this work is to construct new feedback functions with optimal periods for large values of n . This work would be conducted in several phases as follows:

- ***Extended study and literature survey for NLFSR:*** where the importance of *NLFSRs* and their security strength in recent applications are studied and compared to the *LFSRs*.
- ***Implementation of NLFSR period verifying system:*** where an *NLFSR* period testing algorithm is designed, and a verifier program is implemented to test a list of feedback functions and to find their periods.
- ***Reviewing and verifying the existing functions of various register sizes, n:*** where the existing feedback functions of various register sizes, n , that provide optimal periods for *NLFSRs* are verified. This phase also helps verifying the correctness of the implementation in the previous phase.

- *Proposing new functions with optimal periods for large values of n :*

where new feedback functions with optimal periods for large values of n are constructed (up to $n = 25$).

1.3 Motivations

FSR is one kind of implementation which has been widely used in key stream generators for many stream ciphers cryptosystems and pseudorandom number generators for many cryptographic algorithms. Numerous network security applications based on cryptographic algorithms make use of random numbers to remain secure. There are many applications of pseudorandom numbers, like generation of keys for public key cryptography and digital signature algorithms, generation of a symmetric key for use as a temporary key which is used in several network applications such as TLS protocol, Wi-Fi, E-mail security and IP security. Moreover, pseudorandom numbers can be used in many key distribution protocols such as Kerberos protocol and initialization values (IVs) for handshaking in order to prevent replay attacks and other types of attacks that target the network protocols [6].

The importance of *NLFSR* comes from its ability to produce a highly secure pseudorandom sequence which is harder to break than the one generated by an *LFSR* of the same size. To determine the structure of *NLFSR*, at least $\theta(2^n)$ sequence bits are needed, where n is the register size [5], while breaking an *LFSR* just needs a sequence of $\theta(n)$ bits. Since still have many open fundamental problems,

including finding a systematic method to construct *NLFSRs* with optimal period, this work is considered one step towards these challenges. This is a challenging problem since there is no mathematical foundation supporting it.

Having a complete list of *NLFSRs* feedback functions with optimal periods of different types for different sizes might help stream cipher system designers moving from *LFSRs* which are less secure-domain to *NLFSRs* which are more secure-domain. In addition, providing a complete list of *NLFSRs* feedback functions will give the designers the variety and the freedom to choose the appropriate *NLFSR* feedback functions based on the environment of cryptographic applications such as stream ciphers and light weight block ciphers to provide high-level security in communication systems. Moreover, we hope that our work might help analyzing feedback functions of maximum-period *NLFSRs* and finding a supporting theory characterizing them.

1.4 Contributions

The contributions made by our work are as follows:

- New feedback functions with optimal periods for large values of n are constructed to include all feedback functions of sizes $n = 4, \dots, 25$.
- A new type of feedback function with optimal period is proposed, and all feedback functions of this type are constructed for $n = 5, \dots, 25$.

1.5 Thesis Outline

The reminder of this work is structured as follows. Chapter 2 discusses the background of our work. Chapter 3 presents the related work that focuses on non-linear feedback shift register functions with optimal periods. Chapter 4 highlights the necessary preliminary information and some related definitions, environmental setup, the proposed construction method and the proof of correctness of our method. Chapter 5 discusses the final results obtained from our work and lists new *NLFSR* feedback functions with optimal periods. Finally, Chapter 6 concludes the thesis with future prospects for the work.

CHAPTER 2

BACKGROUND

This chapter is structured as follows. Section 1 presents a review of the concepts related to feedback shift registers, linear feedback shift registers and non linear feedback shift registers. Section 2 presents some usages of *LFSR* and *NLFSR* for different applications. Section 3 discusses the security of *NLFSRs* and their security strength in recent applications as compared to *LFSRs*.

2.1 Feedback Shift Registers (*FSR*)

Feedback shift register (*FSR*) is an implementation that has the property of randomness. *FSR* has n binary storage cells; each cell can hold a single bit. Cell $i \in \{0, 1, \dots, n-1\}$ is linked with a state variable x_i that shows the current value of a cell i and the feedback function $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$ which reveals the updated value of bit i . The state of an *FSR* can be presented as a vector of values of its state variables $(x_0, x_1, \dots, x_{n-1})$. The *period* of the *FSR* is defined as the length of the longest cyclic output sequence that can be obtained. The

value of the cell 0 determines the output of an *FSR*, while the input of an *FSR* is determined by the value of the cell $n - 1$.

FSR can be implemented in either hardware or software. It exists mainly in two types: Linear Feedback Shift Register and Non- Linear Feedback Shift Register.

2.1.1 Linear Feedback Shift Register (*LFSR*)

One of the most prevalent tools used to generate pseudorandom sequences are Linear Feedback Shift Registers (*LFSRs*). They have various applications such as data compression [7], cryptography [8], error detection and correction [9] and testing [10]. Research on *LFSRs* has been developed since early 1960's [11]. Nowadays, *LFSRs* are considered to be a well-known and mature area and most of *LFSRs* 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 of length $2^n - 1$. Berlekamp-Massey algorithm [12] can be used to generate a given binary sequence of a minimal *LFSR* (The smallest length of *LFSR* that can generate the output sequence). Golomb's postulates [11] have characterized the properties of sequences created by *LFSRs* statistically. Unlike *NLFSR*, The current state in *LFSRs* is a linear function of the previous state [4]. The general structure of an n -bit *LFSR* implementation is shown in Figure 2.1.

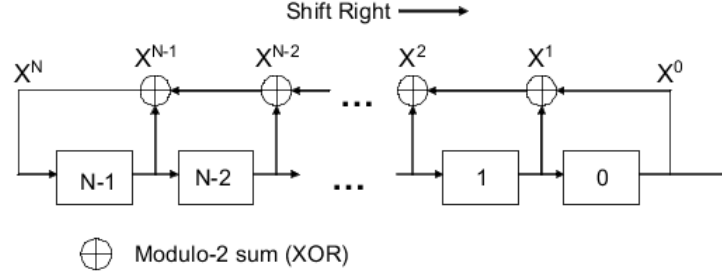


Figure 2.1: The General Structure of an n -bit *LFSR* [2].

A *FSR* is called a Linear Feedback Shift Register (*LFSR*) when it has only linear feedback functions (i.e. of type $f(x_0, x_1, \dots, x_{n-1}) = c_0 \cdot x_0 \oplus c_1 \cdot x_1 \oplus \dots \oplus c_{n-1} \cdot x_{n-1}$, where $c_i \in \{0, 1\}$ for $i \in \{0, 1, \dots, n-1\}$), and $c_0 = 1$ to include the output bit in the feedback. Otherwise, it is called a Nonlinear Feedback Shift Register (*NLFSR*). All terms in *LFSRs* 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 $(c_i \cdot x_j \cdot x_k)$, where $j \neq k$.

In *LFSR* system, there is a unique transformation between the configurations of Galois and Fibonacci. Therefore, It 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).

2.1.2 Non- Linear Feedback Shift Register (*NLFSR*)

The strength of *NLFSR* comes from its ability to produce a very secure pseudo-random sequence which is hard to break (i.e. *NLFSR* needs at least $\theta(2^n)$ bits of a sequence to determine the structure, where n is the register size). *NLFSRs* still have many fundamental challenges, which remain open even with well-understood theory. Getting a systematic method to construct *NLFSRs*, which generates the longest period, might be the most significant issue. There are many algorithms that either deal with particular cases of *NLFSR* [13] [14], or can be implemented only on small values of n (Like $n = 1, 3, 4, 11\text{--}13, 15\text{--}17, 19\text{--}21, 23$) [15]. Generally, the problem is very hard since there is no mathematical foundation supporting it. Figure 2.2 shows the general structure of the feedback function of *NLFSR* with n -bit.

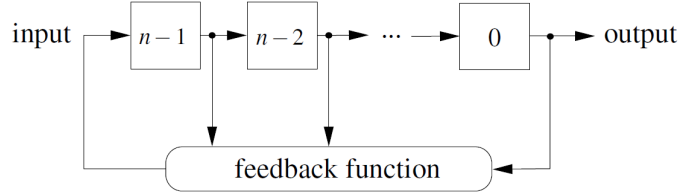


Figure 2.2: The General Structure of an n -bit *NLFSR* [1].

An *NLFSR* can be implemented using either Fibonacci configuration or Galois configuration. The Fibonacci configuration applies the feedback only to the input cell ($n - 1$) of the shift register as shown in Figure 2.2, while the Galois configuration is able to potentially apply the feedback to any cell or stage. An example of a 4-bit Fibonacci *NLFSR* with its feedback function $f(x_0, x_1, x_2, x_3) = x_0 \oplus x_1 \oplus x_2 \oplus x_1 \cdot x_2$ is shown in Figure 2.3.

Unlike *LFSR* system, where the transformation between Fibonacci and Galois *NLFSR* can be done by only adjusting the initial state and reversing the order of feedback taps of *LFSR*, the *NLFSR* system does not have a unique transformation between Fibonacci and Galois configurations [16] [17].

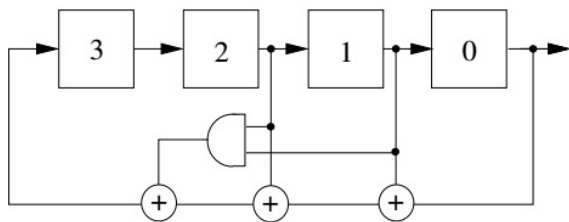


Figure 2.3: A 4-bit Fibonacci *NLFSR* Example [1].

The maximum possible period for an n -bit *NLFSR* is 2^n . The size of period has been extended to be 2^n by adding the complement of a product term $x_1, x_2, \dots, x_n$ to the feedback functions. The complement of x is determined by $\bar{x} = x \oplus 1$.

2.2 Applications of *LFSR* and *NLFSR*

LFSRs and *NLFSRs* can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudorandom sequence, such as direct-sequence spread spectrum radio. They have also been used for generating an approximation of white noise in various programmable sound generators. In this section, we list some usages of *LFSRs* and *NLFSRs* as the following:

- *LFSRs* have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very uniformly distributed output streams. However, an *LFSR* is a linear system, leading to fairly easy cryptanalysis (For more details see section 2.3).
- The repeating sequence of states of an *LFSR* allows it to be used as a clock divider or as a counter when a non-binary sequence is acceptable, as is often the case where computer index or framing locations need to be machine-readable. *LFSR* counters have simpler feedback logic than natural binary counters or Gray-code counters, and therefore can operate at higher clock rates. However, it is necessary to ensure that the *LFSR* never enters an all-zeros state, for example by presetting it at start-up to any other state in the sequence [6].

- Complete *LFSRs* are commonly used as pattern generators for exhaustive testing, since they cover all possible inputs for an n -input circuit. Maximal-length *LFSRs* and weighted *LFSRs* are widely used as pseudorandom test pattern generators for pseudorandom test applications.
- *LFSRs* are used in circuit testing for test pattern generation (for exhaustive testing, pseudorandom testing or pseudo-exhaustive testing) and for signature analysis.
- *LFSRs* are also used in radio jamming systems to generate pseudo-random noise to raise the noise floor of a target communication system [6].
- *NLFSRs* are components in modern stream ciphers, especially in *RFID* and smartcard applications. *NLFSRs* are known to be more resistant to cryptanalytic attacks than *LFSRs*. (For more details see section 2.3).
- *NLFSR* feedback functions are used as building blocks in many recent applications and products of stream cipher cryptosystems such as (Achterbahn, Grain, Mickey, Dragon, Trivium, VEST, ...).

2.3 Security of *NLFSR*

Stream ciphers based on *NLFSR* are very important, especially in environments with high throughput requirements. Although *LFSR* has a large period and linear complexity, it can be reproduced using Berlekamp-Massey algorithm after observing $2n$ output sequence bits [12]. Another drawback of *LFSR* is the linearity problem, that is the same pattern may be repeated on or before completing 2^n potential patterns. Moreover, *LFSRs* are exposed to many attacks like algebraic attack, known plaintext attack and correlation attack [18]. The following points show how LFSR-Based system is not immune to known plaintext attack and how it is easy to break *LFSR* and recover the entire *LFSR* configuration, and thus recover the entire plaintext.

- In Kirchhoff's Principle, we assume that everything is known to the attacker except the key consists of *LFSR* initial fills, that means the attacker know the following:
 - The ciphertext(y_i).
 - The first $(2m)$ of the plaintext $(x_0, \dots, x_{2m-1})$
 - The degree (m) of *LFSR*.
 - Since the attacker knows: $(y_i = x_i + b_i)$, so it can recover $(b_i = y_i + x_i)$, for $(i = 0, 1, 2, \dots, 2m-1)$.

- The goal of the attacker is to recover $(b_{2m}, b_{2m+1}, \dots)$, but there are m unknown variables $(a_0, a_1, \dots, a_{m-1})$. Therefore, the attacker has to solve a system of (m) linear equations with (m) unknown variables (these equations can be easily solved either by Gaussian elimination or matrix inversion).

$$b_{m+1} = a_1 b_m \oplus a_2 b_{m-1} \oplus a_3 b_{m-2} \oplus \dots \oplus a_m b_1$$

$$b_{m+2} = a_1 b_{m+1} \oplus a_2 b_m \oplus a_3 b_{m-1} \oplus \dots \oplus a_m b_2$$

$$b_{m+3} = a_1 b_{m+2} \oplus a_2 b_{m+1} \oplus a_3 b_m \oplus \dots \oplus a_m b_3$$

$$\vdots \qquad \dots$$

$$b_{2m} = a_1 b_{2m-1} \oplus a_2 b_{2m-2} \oplus a_3 b_{2m-3} \oplus \dots \oplus a_m b_m$$

- If the attacker gets *LFSR* initial fills $(a_0, a_1, \dots, a_{m-1})$, he can recover the entire *LFSR* configuration and compute $(b_0, \dots, b_{2m-1}, b_{2m}, b_{2m+1}, \dots)$. Thus, he can decipher $(x_i = y_i + b_i)$, for $(i = 0, 1, 2, \dots)$ and recover the entire plaintext.

NLFSR is the appropriate solution to overcome these drawbacks especially in designing a cryptographically secure stream cipher. However, the security of *NLFSR* is still not well investigated. Grain, one of the well-known *NLFSR* based stream cipher. This stream cipher is vulnerable to key derivation attack¹ and distinguishing attack² [21] [22] [23], and it is shown that part of internal state of Grain stream cipher could be obtained, and therefore it is partially vulnerable to algebraic attack [24]. Trivium is another *NLFSR* based stream cipher, which is vulnerable to many attacks and the initial state of the cipher can be recovered as shown in [25]. Pommerening [26] showed that *NLFSR* is vulnerable to known plaintext attack by knowing one of two numbers; either the number of the period or the number of degrees of the feedback function.

Table 2.1 shows a comparison between *LFSR* and *NLFSR* in terms of the following factors:

- **Maximum Period:** indicates the length of the longest cyclic output sequence which is produced by the shift register [11].
- **Complexity:** is defined as the degree of the minimal generator³ of the output sequence, and it measures the unpredictability⁴ of the sequence. Complexity is an indication for how difficult a sequence might be replicate.

¹The attack by which the attacker can obtain keys of a required format, such as converting the output of Diffie-Hellman key exchange into a specific form of symmetric key or Keyed-hash functions [19].

²The attack by which the attacker can distinguish the random data from the encrypted data by using cryptanalysis techniques on the ciphertext [20].

³The length of the smallest *FSR* that can generate the output sequence.

⁴Unpredictability means that the future elements of the sequences generated by a *FSR* cannot be predicated on the basis of the earlier elements.

- **Algebraic Degree:** of a Boolean function is the number of variables in the largest product-term of its Algebraic Normal Form (*ANF*) [1].
- **Algebraic Attack:** is the attack that solves a system with more than a variable of algebraic equations and breaks any stream cipher especially that has a small number of input state [27] [28].

Table 2.1: A Comparison Between *LFSR* and *NLFSR*

FACTOR	<i>LFSR</i>	<i>NLFSR</i>
Maximum Period	$2^n - 1$	$2^n - 1$
Complexity	$2n$	2^n
Algebraic Degree	1	≥ 2
Algebraic Attack Immunity	<i>NO</i>	<i>YES</i>

CHAPTER 3

RELATED WORK

This chapter presents the existing studies related to non-linear feedback shift register functions with optimal periods. Section 3.1 presents studies that focused mainly on exploring *NLFSR* feedback functions having optimal periods. The studies in this section concentrated on finding new feedback functions that have simple Algebraic Normal Form (ANF) in order to be desirable for many of the security applications such as stream ciphers and lightweight block ciphers to provide security in algorithms communication systems. Section 3.2 presents studies that did not focus mainly on exploring *NLFSR* feedback functions with optimal periods. However, they made use of one or more of *NLFSR* feedback functions as building blocks in many recent applications and products of stream cipher cryptosystems. Finally, section 3.3 summarizes the most related previous studies that focused on exploring *NLFSR* feedback functions and shows a comparison between them.

3.1 Exploring *NLFSR* Feedback Functions

This section presents studies that focused mainly on exploring *NLFSR* feedback functions having optimal periods. The first attempt in order to develop the general theory of (n, k) for *NLFSRs* is presented by Dubrova et.al. [29]. The authors introduced (n, k) -*NLFSRs* as Galois *LFSR* and derived some of their properties. Moreover, the authors recommended their approach as a fast way for generating the output sequence in order to use with stream cipher application.

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.

LFSRs have feedback functions of the following type:

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

where $c_i \in \{0, 1\}$ for $i \in \{0, 1, \dots, n-1\}$). All terms in an LFSR are linear, like $(c_i \cdot x_i)$. However, NLFSRs include nonlinear terms. For example, a NLFSR of degree 2 includes at least one term of the form $(x_i \cdot x_j)$, where $i \neq j$. The primitive *LFSRs* have a drawback as their linear complexity is equal to their order. Berlekamp-Massey algorithm [12] can be used to generate a given binary sequence of a minimal *LFSR*.

Dubrova [1] presented Fibonacci NLFSR feedback functions of degree 2, with optimal periods of $2^n - 1$, for $4 \leq n \leq 24$, in the following three types:

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

where $a, b, c, d, e, h \in \{1, 2, \dots, n-1\}$, $x_i \in \{0, 1\}$, and the addition (XOR) and multiplication (AND) operations are in modulo 2. However, the list in [1] is incomplete, and many feedback functions are missing. Table 3.1 shows a summary of the feedback functions of *NLFSRs* proposed by Dubrova [1] for various values of n ($4 \leq n \leq 24$).

Another heuristic algorithm based approach for finding the period of *NLFSRs* is proposed in [30], which is used to generate a random n -stage boolean full-length shift register sequences with $n \leq 6$. The heuristic algorithm proposed in this study is efficient for 6-argument function, and it is used in constructing n -*NLFSRs* for *FSR-255* family of hash function. However, this approach is only applicable to small *NLFSR* feedback functions.

Table 3.2 outlines the strengths and the weaknesses of the two studies [1] [30] that focused mainly on exploring *NLFSR* feedback functions, and compares them to our work.

Table 3.1: Number of *NLFSR* Feedback Functions for ($4 \leq n \leq 24$) Proposed by Dubrova [1]

n	Type 1	Type 2	Type 3
4	3	2	-
5	4	9	-
6	8	13	-
7	7	24	9
8	15	18	9
9	4	24	12
10	7	28	12
11	3	22	25
12	3	15	18
13	2	19	15
14	1	10	14
15	1	8	15
16	1	6	8
17	-	6	9
18	-	2	5
19	-	3	4
20	-	-	1
21	-	-	4
22	-	2	3
23	-	1	2
24	-	-	1
Total	61	212	166

Table 3.2: The Strengths and the Weaknesses of the Studies Focused on Exploring *NLFSR* Feedback Functions

Reference	Weaknesses	Strengths
[1]	This study is incomplete, and many feedback functions are missing.	- Proposing 3 different types of feedback functions. - Constructing <i>NLFSR</i> functions up to $n = 24$.
[30]	- Only proposing one type of feedback function. - Only applicable for small <i>NLFSRs</i> ($n \leq 6$).	- Efficient for 6-argument feedback function. - Used for <i>FSR</i> – 255 family of hash function.
Our work	- Due to the time limitations, It constructs <i>NLFSR</i> functions up to ($n = 25$).	- Proposing 5 different types of feedback functions. - Applicable for large n.

3.2 Using *NLFSR* Feedback Functions

This section presents studies that did not focus mainly on exploring *NLFSR* feedback functions with optimal periods. However, they proposed few *NLFSR* function examples, for different goals. The studies in this section make use of one or more of *NLFSR* feedback functions as building blocks in many recent applications and products of stream cipher cryptosystems such as (Achterbahn, Grain, Mickey, Dragon, Trivium, VEST, ...).

Rachwalik et.al. [31] implemented *NLFSRs* in Field Programmable Gate Arrays (*FPGA*) in order to design *NLFSRs* with maximum period. This study used exhaustive searching method to explore *NLFSR* functions with optimal periods. Moreover, this study showed some statistical properties of the binary sequences of order of $n = 25$ and 27 .

Dubrova [3] suggested using the Galois configuration of *NLFSRs* to control the circuit complexity of feedback functions easily. This study presented a new idea using the cross-join pairs which are determined by a nonlinear function which is added to the input stage of the *LFSR*. Moreover, this method does not take more time compared to previous methods. Figure 3.1 shows how to use this method to generate keystream for communication systems.

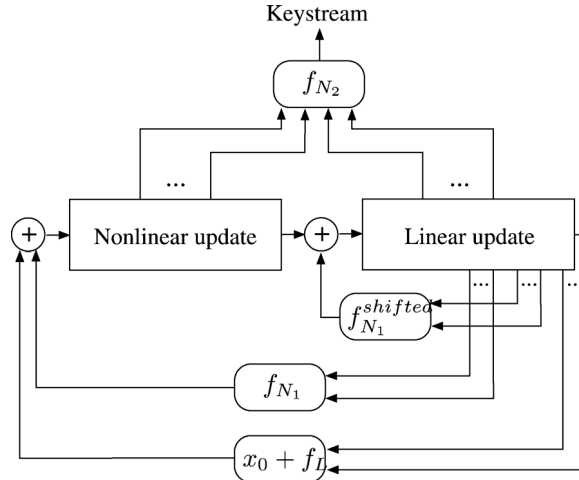


Figure 3.1: An Approach for Generating Keystream Using *NLFSRs* [3].

Another study by Dubrova [32] presented the initial state for *NLFSR* and showed a method to generate the initial state for a Fibonacci *NLFSR* so that it matches a given initial state of the equivalent Galois *NLFSR*. Moreover, Zhiqiang [33] showed the way of transforming a Galois *NLFSR* into the Fibonacci configuration, and the way of mapping between the initial states of them. In addition, this study discussed the problems of the Galois *NLFSRs* and their relation with Golomb's postulates.

Fredricksen [15] showed several algorithms to generate full length *NLFSR* sequences, which are periodic with length equals 2^n , and every two-different binary n -tuples appear exactly one time in a periodic portion of the sequence. The construction of a special class of *NLFSRs* with maximum period has been given by Mykkeltveit et.al. [34] [35]. Recently, maximum period *NLFSRs* for generating de Bruijn sequences of order $n = 23, 24$ and 27 have been constructed, and their software implementations and performances are showed. Moreover, this study used Algebraic Normal Form (ANF) for the software implementation of the three pro-

posed de Bruijn sequences to optimize the computation [36]. G  nther [37] showed how to generate a n -bit *NLFSR* of maximal length 2^n , and the way of generating a De Bruijn sequence by extending a maximal-length *LFSR* with n stages. This approach is considered as a new and very efficient generator of pseudorandom sequences which can be used in stream cipher cryptograph. Sainte-Marie [38] and de Bruijn [39] [40] show that the number of sequences which are cyclic equivalent is given by:

$$B_n = 2^{2^{n-1}-n} \quad (3.1)$$

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 (3.1). 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 (4.4) is given by [31]:

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

Mandal and Gong [41] have defined the relation between a NLFSR and a regular directed graph on a field extension. They assumed two states k^{th} and $(k + 1)^{st}$ such that $S_k = (a_k, a_{k+1}, \dots, a_{k+n-1})$ and $S_{k+1} = (a_{k+1}, a_{k+2}, \dots, a_{k+n})$, 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_{k+1}^i .
- $S_{k+1}^i = (a_{k+1}, a_{k+2}, \dots, a_{n+k-1}, a_{n+k}^i)$, where $a_{n+k}^i \neq a_{n+k}^j$, $i \neq j$, for $i = 1, 2, 3, \dots, 2^q$.

The directed graph (G) here is called de Bruijn graph [39] [42], 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 $j = 1, 2, \dots, 2^q$. The input state is S_k and the output of F is S_{k+1}^i , ($F(S_k) = S_{k+1}^i$) for some values of i which is selected depending on f and Ω where S_{k+1}^i is not generated by F . Let S_0 be an initial state, then the random NLFSR sequence, that F can generate, is given by $a = \{a_0, a_1, \dots, a_k, a_{k+1}, \dots\}$ 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. [43] and Banderier et al. [44] 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 [44].

Liu [13] [45] showed how to use *NLFSR* in 5G mobile applications. This proposed method is a fast alternative to filter generators, and it can be used for applications that need high data rates especially in data communication environments. Dabrowski et.al. [46] proposed and applied a parallel computing technology to find the maximum period *NLFSRs* using exhaustive searching technique.

This proposed method was applied in both hardware and software using simple algebraic normal form. Poluyanenko [47] suggested a new hardware-based approach of *NLFSR* using (*FPGAs*) to generate a maximum cycle sequence with degrees polynomials between 26 and 29. However, this study only proposed 4 *NLFSRs* examples to show the ability of *FPGA* to generate *NLFSR* binary sequences with more efficient asynchronous and parallel architecture than *GPU* and *CPU* microprocessor-based solutions.

An approach suggested by poorghanad et.al. [48] using an evolutionary algorithm to introduce non-linearity. In this approach, the genetic algorithm is used to apply different operations on strings from *LFSR* to enhance their quality in order to meet the criteria of fitness function based on *NIST* standards.

Grain and Trivium are two kinds of *NLFSRs* with time varying feedback functions reported by Hu et.al. [49]. The structure of Grain consists of one *NLFSR* and one *LFSR*. While the structure of Trivium consists of three *NLFSRs*. The authors implemented these two types of *NLFSRs* namely, Grain-like and Trivium-like structures, and presented the results related to their periods.

At the end of this section, we leave the reader with Table 3.3 which outlines the strengths and the weaknesses of the related work discussed in this section. The presented studies in this table make use of one or more of *NLFSR* feedback functions for different security applications.

Table 3.3: The Strengths and the Weaknesses of the Studies Using *NLFSR* Feedback Functions

Reference	Weaknesses	Strengths
[31]	Only proposing 7 <i>NLFSR</i> functions	Minimizing search time of <i>NLFSRs</i> using <i>FPGAs</i>
[36]	Only proposing 3 <i>NLFSR</i> functions	Optimizing the computation using simple <i>ANF</i>
[46]	Only proposing one type of feedback function	<i>NLFSR</i> functions with simple <i>ANF</i>
[47]	Only proposing 4 <i>NLFSR</i> functions	Minimizing search time of <i>NLFSRs</i> using <i>FPGAs</i>
[49]	<i>NLFSR</i> functions with complex <i>ANF</i>	Combining the features of Grain & Trivium stream cipher

3.3 Summary

The purpose of this review is to show that many researchers focused on exploring *NLFSR* functions with optimal periods. However, much of their work is either restricted to particular cases, or only applicable to small *NLFSRs*. Section 3.1 presents studies that focused mainly on exploring *NLFSR* feedback functions having optimal periods. While section 3.2 presents studies that did not focus mainly on exploring *NLFSR* feedback functions with optimal periods. However, they proposed few *NLFSR* examples of feedback functions, for different objectives such as presenting the statistical properties of the binary sequences generated by *NLFSR* feedback functions with optimal periods, showing the capability of generating the random numbers in many applications like *RFID* tags and showing the ability of

FPGA to generate *NLFSR* binary sequences with more efficient asynchronous and parallel architecture than *GPU* and *CPU* microprocessor-based solutions.

Table 3.4 summarizes the most related studies that focused on exploring *NLFSR* functions with optimal periods, and compares them to our work. The comparison is done in terms of three main factors: the method to search for *NLFSR* feedback functions with optimal periods, the value of n (*the size of the shift register*) and the completeness check of finding *NLFSR* feedback functions. We note that these studies are either restricted to particular cases, or only applicable to small *NLFSRs*. In this work, we close the gap by constructing the missing feedback functions with optimal periods in the previous studies. Moreover, we propose a new type of feedback function that also provides optimal periods for *NLFSRs*. We hope that our work might help analyzing feedback functions of maximum-period *NLFSRs* and finding a supporting theory characterizing them.

Table 3.4: Summary of Existing Studies on *NLFSR* Functions with Optimal Periods

Year and Reference	Method	Value of n	Completeness Check
2004 [30]	Heuristic Algorithm for a Random Generation	$n \leq 6$	All the possible <i>NLFSR</i> Functions
2012 [36]	Exhaustive Searching	$n = 23, 24, 27$	Special Case ¹
2012 [1]	Exhaustive Searching	$n \leq 24$	Most of the <i>NLFSR</i> Functions
2012 [31]	Hardware-Based Approach Using <i>FPGA</i>	$n = 25, 27$	Special Case ²
2017 [47]	Hardware-Based Approach Using <i>FPGA</i>	$26 \leq n \leq 29$	Special Case ³
Our Work	Exhaustive Searching	$n \leq 25$	All the possible <i>NLFSR</i> Functions

¹ Only 3 *NLFSRs* examples are proposed in order to generate the longest de Bruijn sequences to show the capability of generating the random numbers in many applications such as *RFID* tags.

² Only 7 *NLFSRs* examples of algebraic degree three and four are proposed to show the statistical properties of *NLFSRs*.

³ Only 4 *NLFSRs* examples are proposed to show the ability of *FPGA* to generate *NLFSR* binary sequences with more efficient asynchronous and parallel architecture than *GPU* and *CPU* microprocessor-based solutions.

CHAPTER 4

METHODOLOGY

In this chapter, section 1 introduces the concept of feedback shift registers and related definitions. Section 2 is dedicated to describe the proposed construction method (Sequential *NLFSR* Function Construction Method), while the proof of correctness of our method is covered in Section 3.

4.1 Preliminaries

This section presents the basic concepts and definitions related to feedback shift registers. 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 $i \in \{0, 1, \dots, n-1\}$ holds a single bit x_i in the state register. The feedback function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ computes a feedback bit used as an input to the register to update the state using a shift operation.

The state of an *FSR* is presented by a vector of bits $X = (x_0, x_1, \dots, x_{n-1})$. 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*.

Definition 1: The *period* of the *FSR* is defined as the length of the longest unrepeated output sequence. The period is called *optimal* if it is equal to $2^n - 1$, where n is the size of the shift register. Due to the limitation of register size, n , there will be a repeated state eventually (when all the 2^n states are exhausted).

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:

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

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

Definition 2: 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 $s = (s_0, s_1, \dots)$ 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:

$$F : (x_0, x_1, \dots, x_{n-1}) \rightarrow ((x_1, \dots, x_{n-1}, f(x_0, x_1, \dots, x_{n-1})))$$

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* [31].

Definition 3: The sequence of de Bruijn $(a_0, \dots, a_{2^n-1})$ of order n which consists of elements from $\mathbb{F}_2$ has a period of 2^n where each n -tuples appears exactly once.

Gammel et al. [50] called the sequences that satisfied the Definition 3, the primitive sequences. In *LFSRs*, these sequences can be produced using primitive polynomials and the theory of such sequences is well-understood [51].

Definition 4: An n -bit *NLFSR* is called a primitive *NLFSR* if its nonlinear feedback function generates a sequence of period $2^n - 1$.

4.2 Environmental Setup

Because our experiments are very computationally expensive and would incur extremely long times if run on standard computers, we use the High Performance Computing Environment (HPC) at the Information Technology Center (ITC) of KFUPM to carry out our experiments especially for large values of n . Table 4.1 shows the technical details and the software used for our experiments. For more detailed specifications about HPC cluster resources, you can visit the following website: <http://hpc.kfupm.edu.sa/Resources.html>.

Table 4.1: Experimental Setup

Number of Nodes	32 Nodes
Processors	Xeon E5-2680 10-Core, 2.8 GHz (Dual-processor)
Number of Cores	660 Cores
Memory	64 GB RAM on each node
Number of Nodes with GPU	12 Nodes
Operating System	Red Hat Enterprise Linux 6
Interconnect	InfiniBand QDR 40Gb/s

4.3 Proposed Construction Method

This work proposes and uses a sequential *NLFSR* function construction method, which sequentially enumerates all feedback functions of a desired type. Then, we 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 computation power since the number of functions grows exponentially in terms of n .

The proposed construction method consists of two parts:

- **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 , SFG sequentially increments the indices of the state 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 types of feedback functions, all other indices are chosen relatively to the 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, 3 and 4) is assigned to $a + 1$ and is 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.

- **Period-Testing Algorithm:** 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 section 4.4 that the seed must appear again in the register since it generates a cycle of periodic states as shown in Figure 4.1. The length of the generated cycle is the period. We argue that the seed cannot be outside the state cycle as shown in Figure 4.2. For a given value of n , if the seed X_0 appears after $2^n - 1$ shift operations, then the maximum period for 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. To shed more light on the scenario shown in Figure 4.1, Algorithm 1 shows how to evaluate the period of a given feedback function with certain parameters.

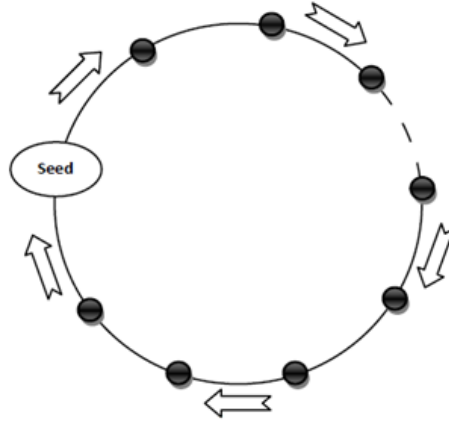


Figure 4.1: The Seed Belongs to the Cycle.

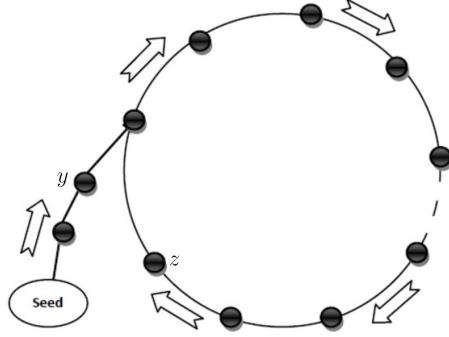


Figure 4.2: The Seed is Outside the Cycle.

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

Algorithm 1 GetPeriod

Input: Initial Seed S , $NLFSR$ f

Output : $period$

```

1:  $T := S$ 
2:  $period := 1$ 
3: while  $True$  do
4:    $T = T[1, \dots, n - 1] + f(T)$ 
5:   if  $S = T$  then
6:      $break$ 
7:   end if
8:    $period = period + 1$ 
9: end while
10: return  $period$ 

```

This work mainly extends Dubrova's work in [1] by proposing a complete list of all new *NLFSR* functions with optimal periods of $2^n - 1$ for all values of $n = 4, 5, \dots, 25$. This work considers the following four types of feedback functions of which the first three types are of degree 2, while the fourth one is of degree 3. These types are the only ones that generate primitive *NLFSRs* for which the algebraic normal form of the Boolean function g in the following formula is quite simple.

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

In addition, these four types of feedback functions have low algebraic degree (2 and 3) and a possibly small number of terms which made them desirable for many cryptographic applications [1]. Moreover, we propose a fifth type of feedback function of degree 3 defined in (4.5) that generates many *NLFSR* feedback functions with optimal periods of $2^n - 1$.

These five types are defined as follows:

$$f_1(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \cdot x_d \quad (4.1)$$

$$f_2(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \cdot x_c \oplus x_d \cdot x_e \quad (4.2)$$

$$f_3(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \oplus x_d \oplus x_e \cdot x_h \quad (4.3)$$

$$f_4(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \oplus x_c \cdot x_d \cdot x_e \quad (4.4)$$

$$f_5(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \cdot x_c \oplus x_d \cdot x_e \cdot x_h \quad (4.5)$$

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

4.4 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. This work examines two scenarios: (Scenario 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 (Scenario B) the seed generates a number of states outside the cycle, then it generates p states in a cycle that excludes the seed. Figures 4.1 and 4.2 illustrate the two Scenarios A and B respectively.

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 4.1 *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 (See Figure 4.2). 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, y and z generate the same output sequence of bits if and only if $y = z$. ■

CHAPTER 5

PROPOSED *NLFSR* FUNCTIONS WITH OPTIMAL PERIODS

This chapter presents the new feedback functions with optimal periods for $n \leq 25$. All the feedback functions listed in [1] for different values of $n \leq 24$ are verified using our period-testing algorithm which was introduced in section 4.3, and it is found that all functions yield optimal periods. However, the list in [1] is incomplete, and many feedback functions are missing. This work constructs the missing feedback functions for $n \leq 25$ for the four types of feedback functions proposed in [1], and proposes a new type of feedback functions with optimal periods for $n \leq 25$ to ensure these functions are the only ones existing within this range.

About 110 experiments for all the five types of feedback functions considered in this work have been conducted (i.e. 22 experiments for each). A total of (1718) new *NLFSR* feedback functions with optimal periods have been yielded. Table 5.1 summarizes the improvement achieved by our construction method in each type of

the feedback functions considered in this work, while Table 5.2 shows a summary of the new feedback functions of *NLFSRs* with optimal periods for various values of n ($4 \leq n \leq 25$). We note that as the value of n (*the size of the shift register*) grows, the probability of getting primitive *NLFSR* feedback functions becomes smaller, as the case of ($n = 25$), where no new functions were found for the first three types of feedback functions, whereas only 1 and 3 feedback functions were found for type 4 and type 5, respectively. Going for larger function sizes seems to be not promising since the time complexity grows exponentially.

Table 5.1: Number of *NLFSR* Feedback Functions for ($4 \leq n \leq 25$)

Type	Proposed	Existing [1]	Total
1	59	61	120
2	212	212	424
3	169	166	335
4	140	127	267
5	1138	-	1138

Table 5.2: Summary of the New Feedback Functions of *NLFSRs* with Optimal Periods for ($4 \leq n \leq 25$)

n	f_1	f_2	f_3	f_4	f_5	n	f_1	f_2	f_3	f_4	f_5
4	3	2	-	-	-	15	1	8	5	4	58
5	4	9	-	4	18	16	1	6	8	3	44
6	8	13	8	8	25	17	-	6	5	1	46
7	7	24	9	11	87	18	-	2	5	3	18
8	15	18	13	22	104	19	-	3	4	3	31
9	4	24	12	14	109	20	-	-	2	2	13
10	7	28	18	13	105	21	-	-	4	-	13
11	3	22	25	15	161	22	-	2	3	1	8
12	3	15	18	17	87	23	-	1	2	-	3
13	2	19	13	7	123	24	-	-	1	-	2
14	1	10	14	11	80	25	-	-	-	1	3
Total							59	212	169	140	1138

5.1 Constructing New *NLFSR* Feedback Functions

In this section, Dubrova's work [1] is extended by constructing all missing feedback functions with optimal periods for all $n = 4, 5, \dots, 25$. These functions are pairwise nonequivalent and different from the existing ones. Tables [5.3–5.6] show examples of the new feedback functions with optimal periods for the values of n in the range ($4 \leq n \leq 25$) for the first four types presented in section 4.3. For the complete list, Tables [7.1–7.22] in the Appendix list all the new feedback functions with optimal periods for the functions of types 1, 2, 3 and 4 in the range ($4 \leq n \leq 25$) (see the Appendix). Any missing entry in these tables implies that no new feedback functions with optimal periods were found for that particular type and value of n .

5.1.1 Proposed Functions of Type 1

Our construction method found 59 new functions with optimal periods for $n = 4 \dots 16$. However, no new functions were found for $17 \leq n \leq 25$. Examples of the new feedback functions of type 1 that have optimal periods are listed in Table 5.3 (see the Appendix for the complete list). Each function in Table 5.3 is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables in type 1 functions, as defined in (4.1) in section 4.3. Notice that e and h are ignored in type 1, and indicated by *N/A* (Not Applicable).

Table 5.3: Examples of Type 1 Functions with Optimal Periods

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
4	0	2	3	1	2	N/A	N/A	11	0	2	10	7	10	N/A	N/A
5	0	2	4	1	4	N/A	N/A	12	0	4	9	3	9	N/A	N/A
6	0	2	5	2	5	N/A	N/A	13	0	2	12	4	8	N/A	N/A
7	0	2	6	1	3	N/A	N/A	14	0	12	13	2	7	N/A	N/A
8	0	2	7	1	7	N/A	N/A	15	0	6	10	4	13	N/A	N/A
9	0	3	8	1	5	N/A	N/A	16	0	3	14	13	14	N/A	N/A
10	0	2	9	3	4	N/A	N/A								

In Table 5.3, the indices of the state variables of *NLFSR* feedback functions are expressed as x_0, x_a, x_b, x_c, x_d , respectively. For example: $f(x_0, x_1, x_2, x_3, x_4) = x_0 \oplus x_2 \oplus x_3 \oplus x_1 \cdot x_2$ is a function of type 1 with register size $n = 4$, and is represented in Table 5.3 by the following entry: [4 0 2 3 1 2 N/A N/A], as shown in Figure 5.1. It should be noted that number of variables in the largest product term (last term) of the presented functions in this table, is equal two. Therefore, this type of feedback function is of algebraic degree 2 which using simple algebraic normal form. This type of functions is recommended for stream ciphers which require lightweight devices.

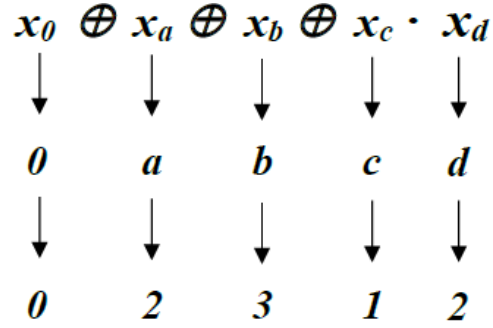


Figure 5.1: An Example of Feedback Function of Type 1 .

Figure 5.2 shows the total number of feedback functions of type 1 obtained in this work for each function size n . For $n = 8$, we obtained a total of 30 feedback functions (proposed and existing) of this type which is the largest among other values of n . We found new feedback functions as well for almost every size of n between 4 and 16. It also shows that there were no existing feedback functions for $n > 16$. However, Dubrova [1] found two functions of size $n = 19$. Moreover, we notice that there are no more new functions that can be obtained for $n = 20, 21, 22, 23, 24$, or 25. Therefore, we conclude that going beyond $n = 25$ seems not promising from the data provided in this figure, and because this process will require long searching time since the time complexity of this process is exponential in the length of the shift register, n .

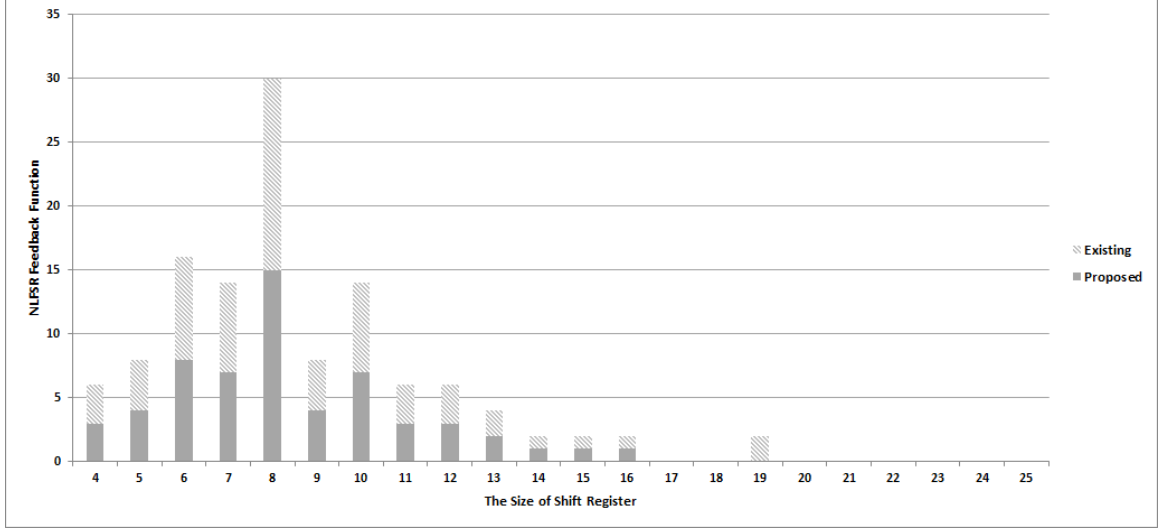


Figure 5.2: Total Number of Feedback Functions (Proposed and Existing) of Type 1.

5.1.2 Proposed Functions of Type 2

For type 2 functions, we proposed 212 new feedback functions with optimal periods for $n = 4 \dots 25$. However, no new functions were found for $n = 20, 21, 24$ or 25 . Examples of the proposed feedback functions of type 2 that have optimal periods are listed in Table 5.4 (see the Appendix for the complete list). Each function in Table 5.4 is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables in type 2 functions, as defined in (4.2) in section 4.3. Notice that h is ignored in type 2, and indicated by *N/A* (Not Applicable).

Table 5.4: Examples of Type 2 Functions with Optimal Periods

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
4	0	2	1	3	2	3	N/A	13	0	7	1	11	8	12	N/A
5	0	3	1	2	3	4	N/A	14	0	8	1	2	5	9	N/A
6	0	3	2	3	2	5	N/A	15	0	8	2	5	4	13	N/A
7	0	4	1	4	3	6	N/A	16	0	9	2	8	4	5	N/A
8	0	5	1	2	4	6	N/A	17	0	9	1	6	5	7	N/A
9	0	5	1	4	6	7	N/A	18	0	15	7	16	11	12	N/A
10	0	6	1	3	2	3	N/A	19	0	13	1	2	11	15	N/A
11	0	6	2	5	7	10	N/A	22	0	17	8	15	10	18	N/A
12	0	9	1	7	2	6	N/A	23	0	20	4	5	4	10	N/A

In Table 5.4, the indices of the state variables of *NLFSR* feedback functions are expressed as $x_0, x_a, x_b, x_c, x_d, x_e$, respectively. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5) = x_0 \oplus x_2 \oplus x_1 \cdot x_3 \oplus x_2 \cdot x_3$ is a function of type 2 with register size $n = 4$, and is represented in Table 5.4 by the following entry: [4 0 2 1 3 2 3 N/A]. The number of variables in the largest product term (last term) of the presented functions in this table, is equal two. Therefore, this type of feedback function is of algebraic degree 2 with fairly simple algebraic normal form. This type of functions might be useful in low-power sensing devices in the Internet of Things (IoT).

The total number of feedback functions of type 2 obtained in our work for each function size n is shown in Figure 5.3. It should be noted that for $n = 10$, a total of 56 feedback functions (proposed and existing) were obtained, which is the largest among other values of n ; whereas for $n = 7$ and 9, they are the second largest among other values with a total of 48 feedback functions for each. In addition, It shows that there were new feedback functions for every size of n between 4 and 19. Moreover, we notice that there were no existing feedback functions for $n \geq 20$. However, we found new functions of size $n = 22$ and 23.

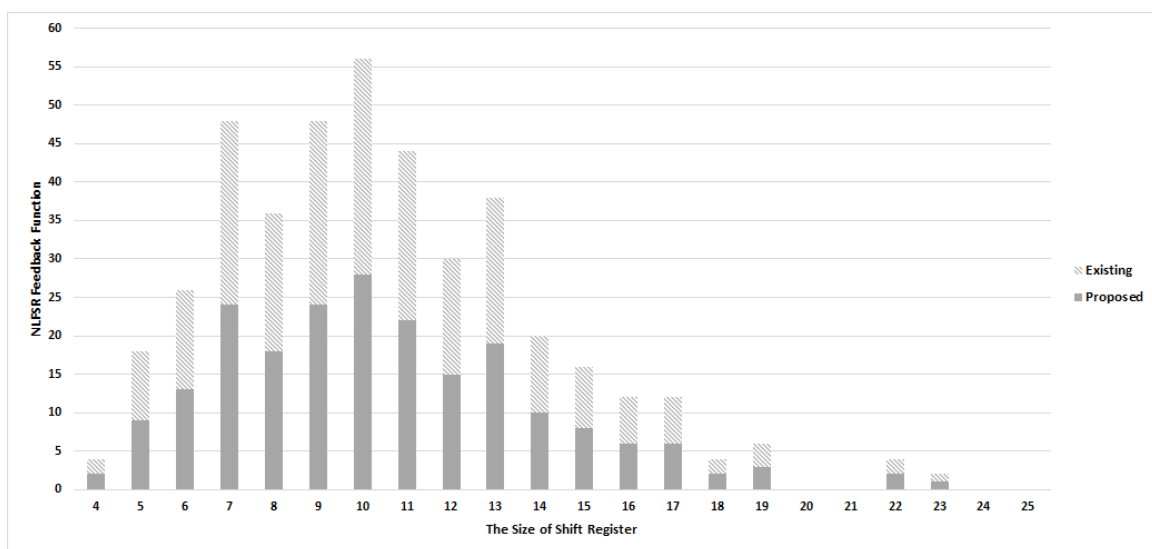


Figure 5.3: Total Number of Feedback Functions (Proposed and Existing) of Type 2.

5.1.3 Proposed Functions of Type 3

We proposed 169 new functions of type 3 with optimal periods for $n = 6 \dots 25$.

There are no new functions for $n = 4, 5$ or 25 . Examples of the new feedback functions of type 3 that have optimal periods are listed in Table 5.5 (see the Appendix for the complete list). Each function in Table 5.5 is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables in type 3 functions, as defined in (4.3) in section 4.3.

Table 5.5: Examples of Type 3 Functions with Optimal Periods

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
6	0	1	2	3	4	1	5	16	0	2	3	11	15	1	2
7	0	1	4	5	6	1	5	17	0	3	5	14	16	7	15
8	0	1	3	6	7	4	6	18	0	3	6	12	16	3	7
9	0	1	4	6	8	4	6	19	0	2	11	15	18	6	18
10	0	1	5	6	9	1	5	20	0	5	13	14	16	13	17
11	0	1	4	7	10	2	10	21	0	2	4	6	20	6	8
12	0	1	6	7	11	4	5	22	0	4	6	12	21	1	19
13	0	2	3	6	12	7	11	23	0	5	7	12	20	4	19
14	0	1	9	10	13	8	13	24	0	9	15	16	23	6	17
15	0	2	6	12	13	8	12								

In Table 5.5, the indices of the state variables of *NLFSR* feedback functions are expressed as $x_0, x_a, x_b, x_c, x_d, x_e, x_h$, respectively. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_1 \cdot x_5$ is a function of type 3 with register size $n = 6$, and is represented in Table 5.5 by the following entry: $[6 \ 0 \ 1 \ 2 \ 3 \ 4 \ 1 \ 5]$. We note that number of variables in the largest product term (last term) of the presented functions in this table, is equal two. Therefore, this type of feedback function is of algebraic degree 2. Comparing with type 1 and 2 functions, number of terms and the circuit complexity of this type is increased. Thus, the usage of this type of functions is somehow limited for some special cases, where the time issue does not matter.

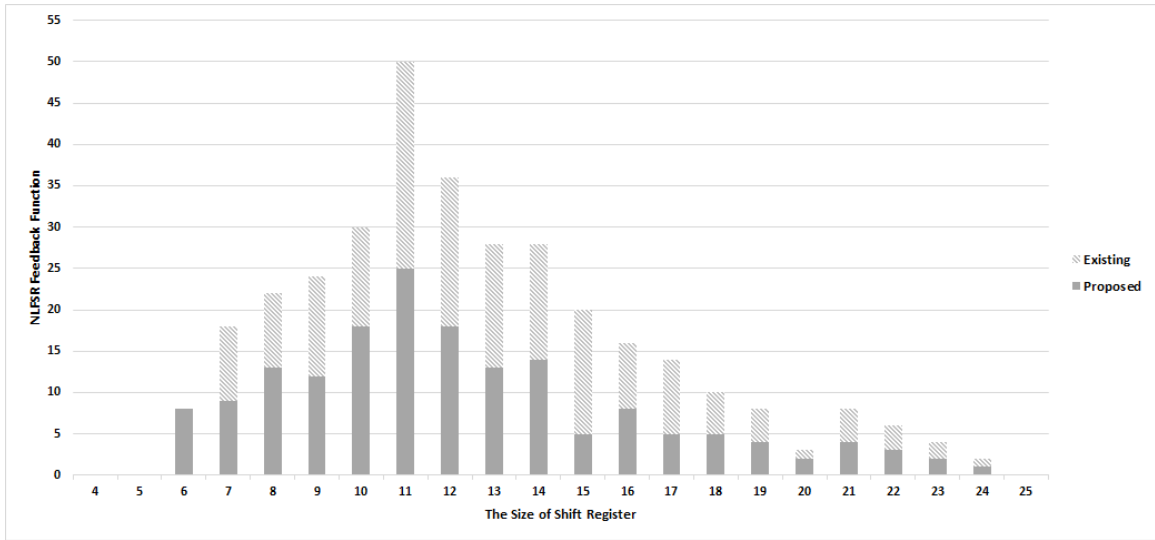


Figure 5.4: Total Number of Feedback Functions (Proposed and Existing) of Type 3.

Figure 5.4 illustrates the total number of feedback functions for type 3 obtained in our work for different sizes of n . We obtained a total of 50 feedback functions of type 3 for $n = 11$, and this is the largest among all different values of n for the shift register. Moreover, we can observe that whenever the size of n between 6 and 24, we found new *NLFSR* feedback functions. It also shows that there were no existing feedback functions for $n = 4$ or 5. However, we found 8 new functions of size $n = 6$. In addition, we also found 13, 18 new functions for $n = 8$ and 10, respectively as proposed in this work. Moreover, we notice that there are no more new functions that can be obtained for $n = 25$.

5.1.4 Proposed Functions of Type 4

For type 4 functions, we proposed 140 new feedback functions with optimal periods for $n = 5 \dots 25$. However, no new functions were found for $n = 4, 21, 23$ or 24. Examples of the proposed feedback functions of type 4 that have optimal periods are listed in Table 5.6 (see the Appendix for the complete list). Each function in Table 5.6 is of the form: $[n \quad 0 \quad a \quad b \quad c \quad d \quad e \quad h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables in type 4 functions, as defined in (4.4) in section 4.3. Notice that h is ignored in type 4, and indicated by *N/A* (Not Applicable).

Table 5.6: Examples of Type 4 Functions with Optimal Periods

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
5	0	2	4	1	2	3	N/A	14	0	2	13	3	7	9	N/A
6	0	3	4	1	3	5	N/A	15	0	5	13	8	9	10	N/A
7	0	2	6	1	3	5	N/A	16	0	8	9	2	4	11	N/A
8	0	2	7	1	2	7	N/A	17	0	10	13	2	14	16	N/A
9	0	3	7	2	3	8	N/A	18	0	9	16	1	12	13	N/A
10	0	2	9	1	2	5	N/A	19	0	8	13	2	10	14	N/A
11	0	2	10	3	8	10	N/A	20	0	6	15	9	10	17	N/A
12	0	3	10	1	5	6	N/A	22	0	16	17	12	17	20	N/A
13	0	5	11	1	3	5	N/A	25	0	9	21	2	4	24	N/A

In Table 5.6, the indices of the state variables of *NLFSR* feedback functions are expressed as $x_0, x_a, x_b, x_c, x_d, x_e$, respectively. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5) = x_0 \oplus x_2 \oplus x_4 \oplus x_1 \cdot x_2 \cdot x_3$ is a function of type 4 with register size $n = 5$, and is represented in Table 5.6 by the following entry: [5 0 2 4 1 2 3 N/A]. The number of variables in the largest product term (last term) of the presented functions in this table, is equal three. Therefore, this type of feedback function is of algebraic degree 3 with simple algebraic normal form. This type of functions is closed to type 1, and it can be useful in limited-power environments.

The total number of feedback functions of type 4 obtained in our work for

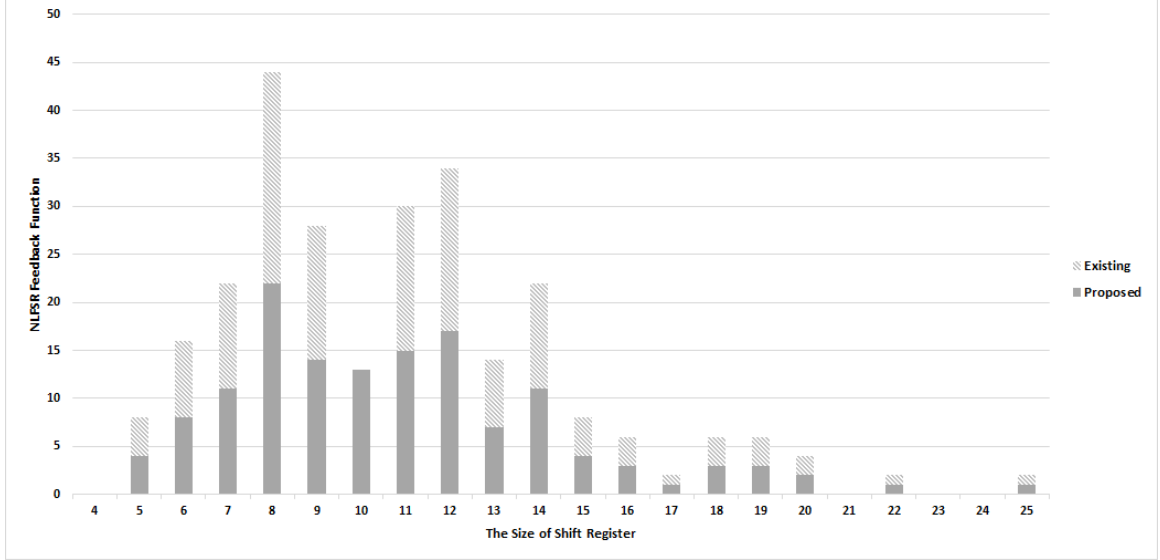


Figure 5.5: Total Number of Feedback Functions (Proposed and Existing) of Type 4.

each function size n is shown in Figure 5.5. We found 13 new functions of size $n = 10$ as proposed in this work. It should be noted that when the size of the register $n = 11$, a total of 44 feedback functions (proposed and existing) of this type were obtained, which is the largest among other values of n . It also shows that there were new feedback functions for every size of n between 4 and 25. However, we notice that there were no existing feedback functions for $n = 4, 21, 23$, or 24 . Thus, from the data provided in this figure, we can note that going for larger function sizes greater than 25 seems to be not promising since the time complexity grows exponentially.

5.2 Proposing New Type of *NLFSR* Feedback Functions

In this section, we tried to examine all possible combinations of the feedback functions of degree 3 with different terms in order to propose new functions of degree 3 that provides optimal periods for *NLFSR*. In particular, the following types of feedback functions are considered:

$$f_5(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \cdot x_c \cdot x_d \quad (5.1)$$

$$f_5(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \cdot x_b \oplus x_c \cdot x_d \cdot x_e \quad (5.2)$$

$$f_5(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \cdot x_b \cdot x_c \oplus x_d \cdot x_e \cdot x_h \quad (5.3)$$

$$f_5(x_0, x_1, \dots, x_{n-1}) = x_0 \oplus x_a \oplus x_b \cdot x_c \oplus x_d \cdot x_e \cdot x_h \quad (5.4)$$

After we verified all the four types of feedback functions using our period testing algorithm, we found that the first three types of feedback functions defined in (5.1), (5.2) and (5.3) did not provide any function with optimal period whereas the last function defined in (5.4) is the only type of feedback function that provides functions with optimal periods for *NLFSRs*. Therefore, we used our construction method (*SFG*) described in section 4.3 to construct all the possible feedback functions of this type (5.4) for $n \leq 25$. Tables [5.8–5.28] list the new feedback functions with optimal periods in the range $(5 \leq n \leq 25)$ for the type defined in (5.4), which is the type 5 considered in this work (see section 5.2.2).

5.2.1 Proposed New Feedback Functions of Type 5

We proposed 1138 functions of type 5 with optimal periods for $n = 5 \dots 25$. There are no new functions for $n = 4$. Examples of the feedback functions of type 5 that have optimal periods are listed in Table 5.7 (see section 5.2.2 for the complete list). Each function in Table 5.7 is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables in type 5 functions, as defined in (5.4) in the previous section.

Table 5.7: Examples of Type 5 Functions with Optimal Periods

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
5	0	1	1	3	1	2	3	16	0	1	2	4	3	4	6
6	0	1	2	3	1	2	3	17	0	1	2	9	3	8	13
7	0	1	1	4	1	3	4	18	0	1	6	7	6	12	16
8	0	1	1	4	2	4	5	19	0	1	2	4	1	3	7
9	0	1	1	6	1	5	6	20	0	1	7	10	3	12	14
10	0	1	2	3	1	2	3	21	0	1	10	14	3	7	11
11	0	1	1	3	3	4	5	22	0	2	5	7	1	13	18
12	0	1	2	5	2	3	11	23	0	7	5	9	8	12	22
13	0	1	1	9	3	4	9	24	0	1	3	6	2	10	14
14	0	1	2	4	2	5	8	25	0	6	10	14	9	13	14
15	0	1	3	9	4	6	8								

In Table 5.7, the indices of the state variables of *NLFSR* feedback functions are expressed as $x_0, x_a, x_b, x_c, x_d, x_e, x_h$, respectively. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3$ is a function of type 5 with register size $n = 5$, and is represented in Table 5.7 by the following entry: [5 0 1 1 3 1 2 3]. We note that number of variables in the largest product term (last term) of the presented functions in this table, is equal three. Therefore, this type of feedback function is of algebraic degree 3. This type of function is close to type 3 in terms of number of the tabs and increasing the circuit complexity of the function. This type of functions is new, and it can meet the purposes of security protection for many cryptographic applications especially increasing the complexity of the modern stream cipher cryptosystems.

Figure 5.6 shows the new *NLFSR* feedback functions of Type 5 which are explored by our construction method for each function size n . We notice that our proposed type (Type 5) is quite promising comparing with other four types proposed by Dubrova [1]. For $n = 11$, we obtained 161 new feedback functions of this type which is the largest among other values of n . It also shows that there were no existing feedback functions for $n = 4$. Moreover, It should be noted that there are only few new feedback functions that can be obtained for $n = 23, 24$ and 25.

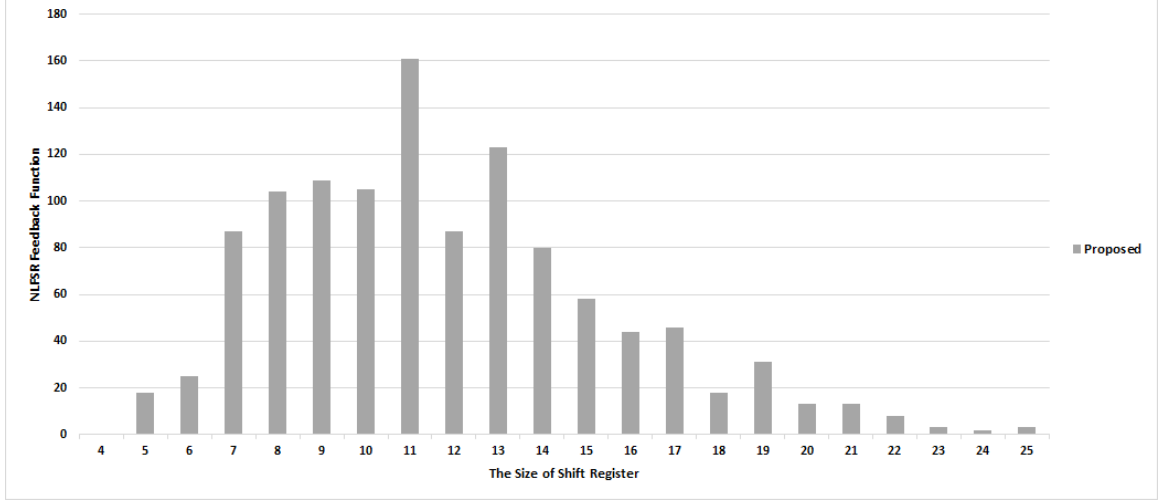


Figure 5.6: Exploring New *NLFSR* Feedback Functions of Type 5.

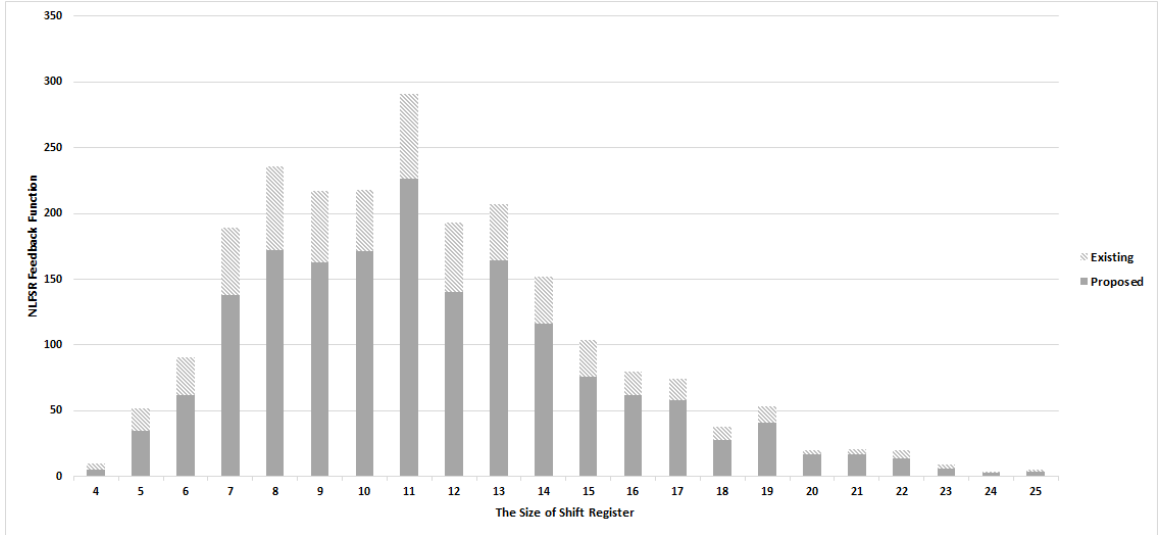


Figure 5.7: Total Number of Feedback Functions (Proposed and Existing) For all Types.

Figure 5.7 shows the total number of feedback functions for all the five types considered in our work. It shows that our construction method proposed new *NLFSR* feedback functions for almost every size of n between 4 and 25. It should be noted that we proposed a large number of *NLFSR* feedback functions comparing with the feedback functions proposed by Dubrova [1]. Moreover, For $n = 11$, we obtained a total of 291 feedback functions for all types which is the largest

among other values of n . Since there were no new feedback functions for $n = 25$ for the first three types of feedback functions, whereas only 1 and 3 feedback functions were found for type 4 and type 5, respectively. Moreover, because feedback functions enumeration process requires long searching time since the time complexity of this process is exponential in the length of the shift register, n . Therefore, we conclude that going beyond $n = 25$ seems not promising and we decide to stop our experiments of exploring new *NLFSR* feedback functions with $n = 25$.

5.2.2 List of the Proposed Feedback Functions of Type 5

In this section, we reiterate the explanation of the proposed feedback functions of type 5 for completeness and clarity purpose. Tables [5.8–5.28] list the new feedback functions of type 5 with optimal periods for $(5 \leq n \leq 25)$.

Table 5.8 lists the new feedback functions of type 5 that have optimal periods for $n = 5$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_2 \oplus x_1 \cdot x_2 \oplus x_1 \cdot x_2 \cdot x_4$ is a function of type 5 with register size $n = 5$, and is represented in Table 5.8 by the following entry: $[5 \ 0 \ 2 \ 1 \ 2 \ 1 \ 2 \ 4]$. This type of feedback function is of algebraic degree 3 because of the number of variables in the largest product term (last term) of the presented functions in this table is equal three. We obtained (18) new feedback functions of this type for $(n = 5)$ which are constructed by our construction method (*SFG*) described in section 4.3.

Table 5.8: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 5$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
5	0	1	1	3	1	2	3	5	0	2	1	2	1	2	4
5	0	1	1	3	2	3	4	5	0	2	1	2	1	3	4
5	0	1	1	4	1	2	4	5	0	2	1	4	1	2	3
5	0	1	1	4	1	3	4	5	0	2	1	4	1	2	4
Continued on next page															

Table 5.8 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
5	0	1	2	3	1	2	3	5	0	2	1	4	2	3	4
5	0	1	2	3	1	2	4	5	0	2	2	3	1	2	3
5	0	1	2	3	2	3	4	5	0	2	2	3	2	3	4
5	0	1	3	4	1	2	3	5	0	2	3	4	1	2	4
5	0	1	3	4	2	3	4	5	0	2	3	4	1	3	4

Table 5.9 lists the feedback functions of type 5 that have optimal periods for $n = 6$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3$ is a function of type 5 with register size $n = 6$, and is represented in Table 5.9 by the following entry: $[5 \ 0 \ 1 \ 2 \ 3 \ 1 \ 2 \ 3]$. A total of (25) new feedback functions for ($n = 6$) have been yielded.

Table 5.9: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 6$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
6	0	1	2	3	1	2	3	6	0	2	1	3	1	2	5
6	0	1	2	3	1	2	4	6	0	2	1	3	1	4	5
6	0	1	2	3	1	2	5	6	0	2	1	3	2	4	5
6	0	1	2	3	2	3	5	6	0	2	1	3	3	4	5
6	0	1	2	3	3	4	5	6	0	2	1	4	1	3	5
6	0	1	2	4	1	2	3	6	0	2	1	5	1	3	4
6	0	1	2	4	1	2	4	6	0	2	1	5	1	3	5
Continued on next page															

Table 5.9 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
6	0	1	2	4	1	2	5	6	0	2	3	5	1	3	4
6	0	1	2	5	1	3	4	6	0	2	3	5	2	3	4
6	0	1	3	4	2	3	5	6	0	2	3	5	2	3	5
6	0	1	3	4	3	4	5	6	0	3	1	4	1	2	3
6	0	1	4	5	1	2	3	6	0	3	1	4	3	4	5
6	0	1	4	5	2	4	5								

Table 5.10 lists the feedback functions of type 5 that have optimal periods for $n = 7$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_4 \oplus x_1 \cdot x_3 \cdot x_4$ is a function of type 5 with register size $n = 7$, and is represented in Table 5.10 by the following entry: $[5 \ 0 \ 1 \ 1 \ 4 \ 1 \ 3 \ 4]$. For $n = 7$, we obtained (87) new feedback functions with optimal periods of this type.

Table 5.10: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 7$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
7	0	1	1	4	1	3	4	7	0	1	3	5	1	3	5
7	0	1	1	4	2	3	4	7	0	1	3	5	3	4	5
7	0	1	1	4	3	5	6	7	0	1	3	5	4	5	6
7	0	1	1	5	1	4	5	7	0	1	4	5	1	2	6
7	0	1	1	6	1	3	6	7	0	1	4	5	1	3	4
7	0	1	1	6	1	5	6	7	0	1	4	5	2	3	5
7	0	1	2	3	1	3	5	7	0	1	4	6	1	3	5
7	0	1	2	3	2	3	5	7	0	1	5	6	1	4	5
7	0	1	2	5	1	2	4	7	0	1	5	6	3	5	6
7	0	1	2	5	1	4	5	7	0	2	1	2	1	2	6
7	0	1	2	5	1	5	6	7	0	2	1	3	1	3	5
Continued on next page															

Table 5.10 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
7	0	1	2	5	2	3	5	7	0	2	1	3	2	4	6
7	0	1	2	5	2	5	6	7	0	2	1	4	1	3	4
7	0	1	3	4	1	3	4	7	0	2	1	4	1	4	6
7	0	1	3	4	1	3	5	7	0	2	1	4	2	4	6
7	0	1	3	4	2	4	5	7	0	2	1	6	1	5	6
7	0	1	3	5	1	2	5	7	0	2	1	6	2	3	5
7	0	2	1	6	2	4	6	7	0	2	3	5	2	4	6
7	0	2	1	6	2	5	6	7	0	2	3	6	1	2	4
7	0	2	1	6	4	5	6	7	0	2	3	6	3	4	6
7	0	2	2	3	3	4	5	7	0	2	4	5	1	2	3
7	0	2	2	3	4	5	6	7	0	2	4	6	1	2	5
7	0	2	2	5	1	2	5	7	0	2	4	6	1	2	6
7	0	2	2	5	2	3	5	7	0	2	4	6	2	5	6
7	0	2	2	5	2	5	6	7	0	3	1	2	1	2	5
7	0	2	2	6	1	3	5	7	0	3	1	3	1	3	5
7	0	2	2	6	1	3	6	7	0	3	1	5	3	5	6
7	0	2	2	6	1	4	6	7	0	3	1	6	1	5	6
7	0	2	2	6	2	4	6	7	0	3	1	6	4	5	6
7	0	2	3	4	1	2	3	7	0	3	2	3	1	4	5
7	0	2	3	4	2	3	5	7	0	3	2	3	2	5	6
7	0	2	3	4	2	4	5	7	0	3	2	4	1	3	5
7	0	2	3	4	2	5	6	7	0	3	2	5	1	3	5
7	0	2	3	4	3	5	6	7	0	3	2	5	1	4	5
7	0	2	3	4	4	5	6	7	0	3	2	5	2	3	4
7	0	3	2	5	2	3	6	7	0	3	4	5	2	3	5
7	0	3	2	5	4	5	6	7	0	3	4	5	2	3	6
7	0	3	3	4	1	3	4	7	0	3	4	5	3	5	6
7	0	3	3	4	3	4	5	7	0	3	5	6	1	5	6
7	0	3	3	4	3	4	6	7	0	3	5	6	2	3	5
7	0	3	3	5	1	4	5	7	0	3	5	6	2	4	5
7	0	3	3	5	3	5	6	7	0	3	5	6	3	4	5
7	0	3	4	5	1	2	5	7	0	3	5	6	3	5	6
7	0	3	4	5	1	4	5								

Table 5.11 lists the feedback functions of type 5 that have optimal periods for $n = 8$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_4 \oplus x_2 \cdot x_4 \cdot x_5$ is a function of type 5 with register size $n = 8$, and is represented in Table 5.11 by the following entry: $[5 \ 0 \ 1 \ 1 \ 4 \ 2 \ 4 \ 5]$. A total of (104) new feedback functions for ($n = 8$) have been constructed.

Table 5.11: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 8$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
8	0	1	1	4	2	4	5	8	0	1	2	6	1	4	7
8	0	1	1	7	1	3	7	8	0	1	2	6	2	3	7
8	0	1	1	7	1	5	7	8	0	1	2	6	2	4	6
8	0	1	2	3	1	4	5	8	0	1	2	7	2	4	6
8	0	1	2	3	2	3	7	8	0	1	3	4	1	2	3
8	0	1	2	4	1	2	3	8	0	1	3	4	1	5	6
8	0	1	2	4	1	3	6	8	0	1	3	4	2	6	7
8	0	1	2	4	1	6	7	8	0	1	3	5	2	4	6
8	0	1	2	4	2	3	6	8	0	1	3	6	1	2	7
8	0	1	2	5	1	2	4	8	0	1	3	6	1	4	7
8	0	1	2	5	1	2	6	8	0	1	3	6	2	4	5
8	0	1	2	5	1	2	7	8	0	1	3	6	2	4	6
8	0	1	2	5	1	5	6	8	0	1	3	6	2	5	7
8	0	1	2	5	2	4	6	8	0	1	3	7	3	5	7
8	0	1	2	5	3	6	7	8	0	1	4	5	3	4	6
8	0	1	2	5	5	6	7	8	0	1	4	6	2	3	7
8	0	1	2	6	1	2	6	8	0	1	4	7	1	6	7
8	0	1	4	7	2	3	4	8	0	2	1	7	2	5	6
8	0	1	5	6	2	3	4	8	0	2	1	7	3	5	7
8	0	1	5	6	2	4	6	8	0	2	2	5	1	5	7
8	0	1	6	7	1	2	6	8	0	2	2	5	3	5	7
8	0	1	6	7	2	4	5	8	0	2	3	5	1	2	7
8	0	1	6	7	2	6	7	8	0	2	3	5	1	5	6
8	0	1	6	7	5	6	7	8	0	2	3	5	2	3	7
8	0	2	1	3	2	3	7	8	0	2	3	7	1	2	4

Continued on next page

Table 5.11 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
8	0	2	1	3	4	6	7	8	0	2	3	7	1	2	6
8	0	2	1	5	1	2	3	8	0	2	3	7	1	3	6
8	0	2	1	5	1	5	6	8	0	2	3	7	2	6	7
8	0	2	1	5	2	3	5	8	0	2	3	7	4	6	7
8	0	2	1	5	3	4	6	8	0	2	4	5	1	5	7
8	0	2	1	5	3	6	7	8	0	2	5	7	1	2	6
8	0	2	1	6	3	5	7	8	0	2	5	7	2	3	6
8	0	2	1	7	1	2	3	8	0	2	5	7	3	6	7
8	0	2	1	7	1	2	6	8	0	2	5	7	4	5	7
8	0	2	1	7	2	3	4	8	0	3	1	4	1	2	6
8	0	3	1	4	1	6	7	8	0	3	4	5	1	2	4
8	0	3	1	4	3	4	6	8	0	3	4	5	1	3	6
8	0	3	1	4	4	6	7	8	0	3	4	5	1	4	6
8	0	3	1	6	2	4	5	8	0	3	4	5	2	3	4
8	0	3	1	7	3	5	7	8	0	3	4	5	3	6	7
8	0	3	2	3	1	2	6	8	0	3	4	6	2	4	6
8	0	3	2	3	2	6	7	8	0	3	4	6	4	5	6
8	0	3	2	4	1	4	7	8	0	3	4	6	4	5	7
8	0	3	2	6	1	2	3	8	0	3	4	7	2	4	5
8	0	3	2	6	1	2	6	8	0	3	4	7	3	4	7
8	0	3	2	6	1	3	5	8	0	3	4	7	4	5	6
8	0	3	2	6	1	4	6	8	0	3	4	7	4	6	7
8	0	3	2	6	1	4	7	8	0	3	5	6	1	2	3
8	0	3	2	6	2	4	6	8	0	3	5	6	3	4	7
8	0	3	2	6	4	6	7	8	0	3	5	7	1	3	7
8	0	3	2	7	1	2	7	8	0	3	6	7	2	4	6
8	0	3	4	5	1	2	3	8	0	3	6	7	3	6	7

Table 5.12 lists the feedback functions of type 5 that have optimal periods for $n = 9$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_6 \oplus x_1 \cdot x_5 \cdot x_6$ is a function of type 5 with register size $n = 9$, and

is represented in Table 5.12 by the following entry: $[5 \ 0 \ 1 \ 1 \ 6 \ 1 \ 5 \ 6]$.

We obtained (109) new feedback functions with optimal periods of this type for $(n = 9)$.

Table 5.12: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 9$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
9	0	1	1	6	1	5	6	9	0	1	3	6	2	4	7
9	0	1	1	6	2	3	7	9	0	1	3	6	4	5	7
9	0	1	2	3	1	3	8	9	0	1	3	8	2	5	7
9	0	1	2	4	1	3	5	9	0	1	4	5	1	7	8
9	0	1	2	5	1	5	7	9	0	1	4	5	3	5	7
9	0	1	2	6	2	4	6	9	0	1	4	5	4	5	8
9	0	1	2	6	4	6	8	9	0	1	4	7	1	4	7
9	0	1	2	7	1	2	7	9	0	1	5	6	1	4	8
9	0	1	3	4	3	4	6	9	0	1	5	6	1	6	8
9	0	1	3	5	1	2	4	9	0	1	5	6	1	7	8
9	0	1	3	5	2	3	6	9	0	1	5	8	1	2	4
9	0	1	3	5	2	4	6	9	0	1	5	8	1	4	7
9	0	1	3	5	3	5	7	9	0	1	5	8	3	4	5
9	0	1	3	6	1	2	4	9	0	1	6	7	1	3	4
9	0	1	3	6	1	3	6	9	0	1	6	8	1	3	7
9	0	1	3	6	1	4	6	9	0	1	6	8	3	5	7
9	0	1	3	6	1	7	8	9	0	1	7	8	1	2	7
9	0	2	1	4	2	3	4	9	0	2	3	8	5	6	8
9	0	2	1	4	3	6	8	9	0	2	3	8	6	7	8
9	0	2	1	6	4	7	8	9	0	2	4	5	1	2	7
9	0	2	1	7	2	4	8	9	0	2	4	5	2	4	5
9	0	2	1	7	2	6	8	9	0	2	4	5	3	4	6
9	0	2	1	8	2	4	7	9	0	2	4	5	4	7	8
9	0	2	2	3	5	6	7	9	0	2	4	6	1	2	8
9	0	2	2	6	1	3	5	9	0	2	4	6	1	3	7
9	0	2	2	7	2	7	8	9	0	2	4	6	2	4	5
9	0	2	3	4	1	2	4	9	0	2	4	6	2	5	7
9	0	2	3	4	1	7	8	9	0	2	4	7	1	2	4
9	0	2	3	4	3	4	5	9	0	2	4	7	2	5	6
9	0	2	3	5	3	5	7	9	0	2	4	7	3	4	8
9	0	2	3	6	1	4	5	9	0	2	4	7	4	5	8
9	0	2	3	6	3	4	5	9	0	2	4	8	1	2	7
Continued on next page															

Table 5.12 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
9	0	2	3	6	3	5	8	9	0	2	4	8	1	2	8
9	0	2	3	8	2	3	5	9	0	2	4	8	1	4	7
9	0	2	3	8	3	4	6	9	0	2	4	8	2	4	8
9	0	2	5	6	1	3	4	9	0	4	2	3	3	4	5
9	0	2	5	6	1	3	8	9	0	4	2	3	3	6	7
9	0	2	5	6	2	3	8	9	0	4	2	3	4	6	7
9	0	2	5	6	3	6	8	9	0	4	2	5	1	6	8
9	0	2	6	8	1	4	6	9	0	4	2	6	1	2	5
9	0	2	6	8	2	6	7	9	0	4	2	6	5	7	8
9	0	2	6	8	3	6	7	9	0	4	2	7	1	6	8
9	0	2	7	8	1	7	8	9	0	4	2	7	6	7	8
9	0	3	1	4	3	4	5	9	0	4	2	8	3	5	8
9	0	3	1	4	4	7	8	9	0	4	3	5	2	4	8
9	0	3	1	6	1	4	7	9	0	4	3	6	2	3	5
9	0	3	1	7	1	6	8	9	0	4	3	8	2	3	7
9	0	3	1	7	2	6	7	9	0	4	3	8	4	5	8
9	0	3	2	5	4	5	8	9	0	4	5	6	3	5	6
9	0	3	4	7	2	3	6	9	0	4	5	8	2	3	7
9	0	3	4	7	3	4	8	9	0	4	5	8	2	5	8
9	0	3	5	8	1	7	8	9	0	4	6	7	5	6	7
9	0	4	2	3	2	7	8	9	0	4	6	8	3	4	5
9	0	4	6	8	3	7	8	9	0	4	7	8	2	4	5
9	0	4	6	8	4	5	6								

Table 5.13 lists the feedback functions of type 5 that have optimal periods for $n = 10$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3$ is a function of type 5 with register size $n = 10$, and is represented in Table 5.13 by the following entry: $[5 \ 0 \ 1 \ 2 \ 3 \ 1 \ 2 \ 3]$. A total of (105) new feedback functions for ($n = 10$) have been proposed.

Table 5.13: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 10$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
10	0	1	2	3	1	2	3	10	0	1	4	7	1	4	7
10	0	1	2	3	1	2	9	10	0	1	4	8	1	4	6
10	0	1	2	4	2	3	7	10	0	1	4	8	1	5	7
10	0	1	2	4	4	6	8	10	0	1	5	6	1	4	5
10	0	1	2	5	2	3	6	10	0	1	5	6	1	4	7
10	0	1	2	5	4	5	8	10	0	1	5	6	2	3	8
10	0	1	2	7	1	4	9	10	0	1	5	7	3	5	9
10	0	1	2	7	2	5	7	10	0	1	6	7	2	3	7
10	0	1	2	8	1	3	7	10	0	1	7	9	5	7	9
10	0	1	2	8	1	7	9	10	0	1	8	9	2	4	5
10	0	1	2	8	3	4	7	10	0	2	1	3	2	4	9
10	0	1	2	9	1	6	8	10	0	2	1	4	1	3	9
10	0	1	3	4	4	7	9	10	0	2	1	4	1	7	9
10	0	1	3	8	2	3	8	10	0	2	1	5	1	6	9
10	0	1	4	5	1	6	9	10	0	2	1	5	2	6	9
10	0	1	4	6	4	7	8	10	0	2	1	7	1	5	8
10	0	1	4	7	1	4	6	10	0	2	1	7	1	8	9
10	0	2	1	7	2	3	5	10	0	3	1	6	6	7	9
10	0	2	1	7	2	6	9	10	0	3	2	4	1	5	8
10	0	2	1	9	1	5	7	10	0	3	2	4	5	6	9
10	0	2	1	9	4	8	9	10	0	3	2	5	2	3	8
10	0	2	3	8	1	3	7	10	0	3	2	6	1	3	7
10	0	2	3	8	1	7	9	10	0	3	2	6	1	8	9
10	0	2	3	9	2	5	9	10	0	3	2	6	2	3	6
10	0	2	3	9	5	7	8	10	0	3	2	6	2	3	8
10	0	2	4	5	1	3	7	10	0	3	2	6	2	5	7
10	0	2	5	7	1	5	9	10	0	3	2	6	2	6	7
10	0	2	5	9	1	4	5	10	0	3	2	6	4	7	8
10	0	2	5	9	4	6	9	10	0	3	2	7	4	7	8
10	0	2	6	7	3	5	9	10	0	3	2	8	1	2	9
10	0	2	7	9	1	7	8	10	0	3	2	8	3	5	6
10	0	3	1	2	2	3	5	10	0	3	2	9	2	3	6
10	0	3	1	2	2	6	8	10	0	3	4	6	2	3	4
10	0	3	1	2	3	6	8	10	0	3	4	6	3	4	5
10	0	3	1	6	4	6	8	10	0	3	4	9	2	6	7
10	0	3	4	9	2	8	9	10	0	4	1	9	1	8	9
10	0	3	4	9	4	6	8	10	0	4	3	5	1	2	8
10	0	3	5	6	1	3	8	10	0	4	3	5	2	5	6
Continued on next page															

Table 5.13 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
10	0	3	5	6	2	8	9	10	0	4	3	5	2	7	9
10	0	3	5	6	3	7	8	10	0	4	3	5	3	7	8
10	0	3	6	7	2	6	8	10	0	4	3	7	1	2	3
10	0	3	6	8	2	4	8	10	0	4	3	7	1	4	6
10	0	3	6	9	2	4	7	10	0	4	3	7	2	3	4
10	0	3	7	8	2	7	8	10	0	4	5	7	1	6	7
10	0	3	8	9	3	8	9	10	0	4	5	7	2	5	9
10	0	3	8	9	5	6	7	10	0	4	5	7	3	4	9
10	0	4	1	5	1	3	9	10	0	4	5	7	7	8	9
10	0	4	1	5	2	5	7	10	0	4	5	9	2	5	6
10	0	4	1	5	3	4	9	10	0	4	6	7	1	7	9
10	0	4	1	5	7	8	9	10	0	4	7	8	1	3	9
10	0	4	1	7	1	5	8	10	0	5	2	7	4	6	9
10	0	4	1	7	4	7	9	10	0	5	2	7	4	7	9
10	0	4	1	7	7	8	9								

Table 5.14 lists the feedback functions of type 5 that have optimal periods for $n = 11$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_3 \oplus x_3 \cdot x_4 \cdot x_5$ is a function of type 5 with register size $n = 11$, and is represented in Table 5.14 by the following entry: $[5 \ 0 \ 1 \ 1 \ 3 \ 3 \ 4 \ 5]$. For $n = 11$, we obtained (161) new feedback functions of this type which is the largest among other values of n .

Table 5.14: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 11$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
11	0	1	1	3	3	4	5	11	0	1	3	9	2	4	9
11	0	1	1	3	8	9	10	11	0	1	3	9	2	7	8
11	0	1	1	6	4	5	6	11	0	1	3	10	3	6	8
11	0	1	1	8	2	3	9	11	0	1	4	5	3	5	7
11	0	1	1	10	1	3	10	11	0	1	4	7	1	3	10
11	0	1	2	3	5	6	7	11	0	1	4	7	1	5	10
11	0	1	2	3	5	7	9	11	0	1	4	7	5	6	9
11	0	1	2	5	3	5	7	11	0	1	4	9	5	6	10
11	0	1	2	7	2	3	4	11	0	1	5	7	3	7	9
11	0	1	2	7	3	5	10	11	0	1	5	8	1	5	8
11	0	1	2	9	2	4	9	11	0	1	5	8	6	9	10
11	0	1	3	4	8	9	10	11	0	1	6	7	1	4	5
11	0	1	3	5	1	4	9	11	0	1	6	7	1	8	10
11	0	1	3	6	3	4	9	11	0	1	6	9	2	5	10
11	0	1	3	6	3	6	8	11	0	1	6	9	3	8	10
11	0	1	3	8	5	8	10	11	0	1	6	9	4	5	6
11	0	1	3	9	1	2	5	11	0	1	6	9	4	9	10
11	0	1	6	10	2	6	10	11	0	2	4	7	4	5	9
11	0	1	7	9	4	5	6	11	0	2	4	9	1	4	9
11	0	1	7	10	3	5	9	11	0	2	4	10	2	6	9
11	0	1	7	10	4	6	7	11	0	2	5	7	1	3	7
11	0	2	1	3	2	4	8	11	0	2	5	10	1	5	10
11	0	2	1	3	4	8	10	11	0	2	5	10	1	8	10
11	0	2	1	4	1	4	10	11	0	2	5	10	2	4	5
11	0	2	1	7	5	7	9	11	0	2	5	10	3	6	8
11	0	2	1	8	2	3	9	11	0	2	6	7	1	5	8
11	0	2	1	8	3	4	10	11	0	2	6	7	3	4	7
11	0	2	1	10	4	6	9	11	0	2	6	9	2	5	8
11	0	2	3	5	2	4	8	11	0	2	6	10	2	5	8
11	0	2	3	5	3	5	7	11	0	2	7	8	3	7	10
11	0	2	3	8	2	4	6	11	0	2	7	8	4	5	8
11	0	2	3	8	2	4	9	11	0	2	7	10	2	8	9
11	0	2	3	8	3	5	6	11	0	2	8	9	3	8	9
11	0	2	4	5	4	6	8	11	0	2	8	10	7	9	10
11	0	2	4	7	1	2	9	11	0	2	9	10	1	2	8
11	0	3	1	4	1	5	8	11	0	3	6	7	2	4	5
11	0	3	1	4	1	7	10	11	0	3	6	7	3	4	7
11	0	3	1	7	2	5	7	11	0	3	6	9	1	2	5

Continued on next page

Table 5.14 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
11	0	3	1	7	2	7	9	11	0	3	7	8	4	5	7
11	0	3	1	8	2	5	10	11	0	3	7	9	1	4	9
11	0	3	1	9	1	4	10	11	0	3	7	9	4	7	8
11	0	3	1	10	5	8	9	11	0	3	7	9	4	9	10
11	0	3	2	7	2	3	7	11	0	3	7	10	1	3	9
11	0	3	2	7	3	7	8	11	0	3	7	10	3	8	10
11	0	3	3	4	7	8	10	11	0	4	1	2	2	3	4
11	0	3	3	8	2	3	8	11	0	4	1	2	4	5	9
11	0	3	3	8	3	7	8	11	0	4	1	2	4	6	9
11	0	3	3	9	3	6	9	11	0	4	1	2	4	6	10
11	0	3	4	7	6	9	10	11	0	4	1	5	3	7	9
11	0	3	4	9	1	2	5	11	0	4	1	6	2	4	7
11	0	3	4	9	1	7	9	11	0	4	1	6	3	5	6
11	0	3	5	6	1	3	10	11	0	4	1	6	4	5	7
11	0	3	5	10	4	5	10	11	0	4	1	7	1	3	9
11	0	4	1	7	3	5	9	11	0	4	6	8	1	4	9
11	0	4	2	5	4	6	7	11	0	4	6	9	4	6	10
11	0	4	2	5	6	8	9	11	0	4	6	9	5	6	8
11	0	4	2	6	1	5	10	11	0	4	6	10	8	9	10
11	0	4	2	6	2	5	7	11	0	4	7	10	2	3	6
11	0	4	2	6	4	6	8	11	0	4	8	9	1	8	10
11	0	4	2	6	5	8	10	11	0	4	8	9	4	6	10
11	0	4	2	8	1	6	7	11	0	4	8	10	7	8	9
11	0	4	2	9	3	6	9	11	0	4	9	10	2	5	9
11	0	4	3	6	1	8	9	11	0	4	9	10	3	4	5
11	0	4	3	8	8	9	10	11	0	5	1	5	2	4	10
11	0	4	3	9	2	6	8	11	0	5	1	5	2	6	10
11	0	4	4	7	2	4	7	11	0	5	1	8	1	7	8
11	0	4	5	6	7	8	9	11	0	5	1	9	2	4	7
11	0	4	5	8	2	7	8	11	0	5	1	10	4	7	8
11	0	4	5	8	4	5	8	11	0	5	1	10	6	7	9
11	0	4	5	8	4	5	10	11	0	5	2	3	2	7	9
11	0	4	6	8	1	2	5	11	0	5	2	3	3	4	5
11	0	5	2	7	5	7	10	11	0	5	6	8	6	8	10
11	0	5	2	8	4	8	10	11	0	5	7	8	3	6	10
11	0	5	3	4	2	3	10	11	0	5	7	9	1	3	5
11	0	5	3	7	1	6	7	11	0	5	7	9	1	4	6
11	0	5	3	7	6	7	8	11	0	5	7	9	2	3	10
11	0	5	3	7	7	9	10	11	0	5	7	9	4	8	9
Continued on next page															

Table 5.14 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
11	0	5	3	8	1	7	10	11	0	5	7	10	5	7	10
11	0	5	4	7	3	7	10	11	0	5	8	9	5	6	9
11	0	5	4	9	4	7	9	11	0	5	9	10	4	5	10
11	0	5	6	7	6	7	9								

Table 5.15 lists the feedback functions of type 5 that have optimal periods for $n = 12$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_5 \oplus x_2 \cdot x_3 \cdot x_{11}$ is a function of type 5 with register size $n = 12$, and is represented in Table 5.15 by the following entry: $[5 \ 0 \ 1 \ 2 \ 5 \ 2 \ 3 \ 11]$. A total of (87) new feedback functions for ($n = 12$) have been yielded.

Table 5.15: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 12$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
12	0	1	2	5	2	3	11	12	0	1	6	7	1	2	6
12	0	1	2	5	2	6	9	12	0	1	6	7	2	8	9
12	0	1	2	5	4	5	6	12	0	1	6	7	5	9	10
12	0	1	2	6	1	2	6	12	0	1	6	8	1	3	6
12	0	1	2	6	4	5	11	12	0	1	6	8	2	3	4
12	0	1	2	6	6	7	8	12	0	1	6	8	5	7	11
12	0	1	2	7	7	8	11	12	0	1	6	9	2	9	10
12	0	1	2	9	4	8	11	12	0	1	7	9	2	6	8
12	0	1	3	4	3	7	10	12	0	1	8	10	3	10	11
12	0	1	3	6	3	5	10	12	0	1	8	11	1	2	6
12	0	1	3	6	6	8	11	12	0	1	8	11	3	6	8
12	0	1	3	10	1	6	10	12	0	1	8	11	6	8	9
12	0	1	3	10	8	9	10	12	0	1	8	11	7	8	11
Continued on next page															

Table 5.15 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
12	0	1	4	8	1	5	11	12	0	1	10	11	9	10	11
12	0	1	4	11	2	4	9	12	0	2	1	3	6	7	8
12	0	1	5	6	2	8	10	12	0	2	1	4	3	5	9
12	0	1	5	7	2	8	10	12	0	2	1	5	1	4	9
12	0	2	1	7	1	2	8	12	0	3	2	5	3	5	8
12	0	2	1	7	2	3	6	12	0	3	2	5	3	7	10
12	0	2	1	7	4	5	10	12	0	3	4	7	5	6	9
12	0	2	1	11	4	5	8	12	0	3	4	10	3	5	11
12	0	2	3	4	1	5	11	12	0	3	4	10	3	8	9
12	0	2	3	7	5	9	10	12	0	3	5	8	1	4	5
12	0	2	3	9	1	6	9	12	0	3	5	8	2	7	9
12	0	2	3	9	4	6	7	12	0	3	7	8	2	5	8
12	0	2	5	9	1	5	9	12	0	4	1	5	4	6	8
12	0	2	6	11	1	3	9	12	0	4	1	5	4	7	8
12	0	2	7	11	4	10	11	12	0	4	5	9	1	2	7
12	0	2	8	9	1	3	5	12	0	4	5	9	4	6	8
12	0	2	9	10	1	3	11	12	0	4	7	11	4	7	8
12	0	2	9	11	3	4	8	12	0	5	1	4	3	4	11
12	0	3	1	4	6	10	11	12	0	5	1	6	1	2	6
12	0	3	1	6	1	4	10	12	0	5	1	10	1	6	10
12	0	3	1	10	7	8	10	12	0	5	2	10	5	7	11
12	0	3	2	5	1	2	6	12	0	5	3	9	1	3	9
12	0	5	4	6	1	2	4	12	0	5	6	9	3	6	9
12	0	5	4	6	1	2	10	12	0	5	6	9	4	6	10
12	0	5	4	6	1	3	7	12	0	5	6	10	1	4	5
12	0	5	4	10	2	3	7	12	0	5	6	10	2	3	6
12	0	5	4	10	2	4	6	12	0	5	6	11	1	6	7
12	0	5	4	10	3	4	6	12	0	5	8	9	3	8	9
12	0	5	6	7	6	7	11	12	0	5	8	10	1	9	10
12	0	5	6	8	1	2	3	12	0	5	8	10	4	10	11
12	0	5	6	8	3	4	10								

Table 5.16 lists the feedback functions of type 5 that have optimal periods for $n = 13$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_1 \cdot x_9 \oplus x_3 \cdot x_4 \cdot x_9$ is a function of type 5 with register size $n = 13$, and is represented in Table 5.16 by the following entry: $[5 \ 0 \ 1 \ 1 \ 9 \ 3 \ 4 \ 9]$. We obtained (123) new feedback functions of this type for ($n = 13$) which are constructed by our construction method (*SFG*) described in section 4.3.

Table 5.16: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 13$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
13	0	1	1	9	3	4	9	13	0	1	6	7	4	8	11
13	0	1	1	11	8	9	12	13	0	1	7	10	1	6	12
13	0	1	2	3	1	4	10	13	0	1	7	11	4	5	6
13	0	1	2	7	1	3	8	13	0	1	7	12	1	3	5
13	0	1	2	7	1	8	12	13	0	1	8	9	1	5	11
13	0	1	2	8	7	9	11	13	0	1	8	9	3	10	11
13	0	1	3	5	5	8	11	13	0	1	9	10	1	10	11
13	0	1	3	7	7	10	11	13	0	1	10	12	1	3	7
13	0	1	3	11	1	2	12	13	0	2	1	3	2	8	10
13	0	1	4	6	5	7	9	13	0	2	1	4	2	11	12
13	0	1	5	6	2	5	9	13	0	2	1	6	2	3	10
13	0	1	5	6	3	11	12	13	0	2	1	10	2	5	8
13	0	1	5	6	4	5	7	13	0	2	3	4	3	4	5
13	0	1	5	7	5	8	10	13	0	2	3	4	5	6	12
13	0	1	5	8	2	5	11	13	0	2	3	10	2	3	12
13	0	1	5	10	1	4	7	13	0	2	3	12	2	5	7
13	0	1	5	11	6	8	11	13	0	2	3	12	3	4	6
13	0	2	4	7	2	6	10	13	0	3	3	10	2	3	10
13	0	2	4	7	5	6	8	13	0	3	5	9	1	3	6
13	0	2	5	8	1	2	10	13	0	3	5	12	2	7	9
13	0	2	6	9	1	7	12	13	0	3	6	9	1	8	10
13	0	2	6	10	2	3	5	13	0	3	6	11	3	6	9
13	0	2	6	12	4	5	12	13	0	3	6	12	5	7	11
Continued on next page															

Table 5.16 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
13	0	2	6	12	5	6	11	13	0	3	7	8	5	9	11
13	0	2	7	10	7	9	10	13	0	3	7	11	2	8	11
13	0	2	8	9	1	5	6	13	0	3	8	10	1	5	9
13	0	2	8	9	1	8	12	13	0	3	9	12	2	3	6
13	0	2	8	11	1	4	9	13	0	3	9	12	2	11	12
13	0	2	8	12	7	9	11	13	0	3	9	12	3	8	12
13	0	2	11	12	1	10	12	13	0	3	11	12	1	3	8
13	0	2	11	12	3	7	12	13	0	4	1	7	4	6	8
13	0	3	1	5	9	11	12	13	0	4	1	12	6	10	11
13	0	3	1	6	1	11	12	13	0	4	2	3	1	2	3
13	0	3	2	6	4	8	12	13	0	4	2	7	3	6	11
13	0	3	2	9	6	9	10	13	0	4	2	10	5	6	7
13	0	4	3	10	4	5	7	13	0	5	6	9	9	10	11
13	0	4	7	9	4	8	12	13	0	5	6	11	4	6	9
13	0	4	7	10	2	7	9	13	0	5	6	11	6	9	10
13	0	4	7	12	1	2	10	13	0	5	7	8	2	9	11
13	0	4	7	12	8	10	12	13	0	5	7	10	2	3	11
13	0	4	8	10	1	2	5	13	0	5	7	12	2	5	11
13	0	4	8	11	2	5	11	13	0	5	7	12	3	10	11
13	0	4	8	11	3	5	10	13	0	5	8	11	1	3	12
13	0	5	1	2	5	8	12	13	0	5	8	11	3	6	12
13	0	5	1	7	2	9	12	13	0	5	9	10	3	4	11
13	0	5	1	9	1	2	5	13	0	5	9	11	1	5	12
13	0	5	1	9	1	5	8	13	0	5	11	12	5	10	11
13	0	5	1	9	2	3	5	13	0	6	2	3	3	5	8
13	0	5	1	9	3	6	9	13	0	6	2	4	3	8	12
13	0	5	1	10	10	11	12	13	0	6	2	6	4	8	11
13	0	5	4	7	9	10	11	13	0	6	2	8	5	6	10
13	0	5	4	9	1	2	6	13	0	6	2	10	1	10	11
13	0	5	4	9	5	6	9	13	0	6	2	11	5	6	10
13	0	6	2	12	1	4	6	13	0	6	4	9	2	5	11
13	0	6	3	4	2	6	10	13	0	6	4	11	3	8	11
13	0	6	3	10	2	3	10	13	0	6	5	8	4	5	6
13	0	6	3	12	1	5	8	13	0	6	6	11	6	11	12
13	0	6	4	5	1	2	10	13	0	6	8	9	1	4	6
13	0	6	4	8	2	9	10	13	0	6	10	12	2	5	6
13	0	6	4	8	3	7	11	13	0	6	10	12	4	6	8
13	0	6	4	8	5	7	12	13	0	6	11	12	1	2	10
13	0	6	4	9	2	3	10								

Table 5.17 lists the feedback functions of type 5 that have optimal periods for $n = 14$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_4 \oplus x_2 \cdot x_5 \cdot x_8$ is a function of type 5 with register size $n = 14$, and is represented in Table 5.17 by the following entry: $[5 \ 0 \ 1 \ 2 \ 4 \ 2 \ 5 \ 8]$. A total of (80) new feedback functions for ($n = 14$) have been constructed.

Table 5.17: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 14$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
14	0	1	2	4	2	5	8	14	0	2	1	7	3	7	11
14	0	1	2	5	1	6	10	14	0	2	1	7	4	9	13
14	0	1	2	9	2	7	8	14	0	2	1	9	1	10	11
14	0	1	2	12	1	8	10	14	0	2	3	5	1	2	6
14	0	1	4	5	2	6	10	14	0	2	3	7	3	7	13
14	0	1	4	7	5	7	12	14	0	2	3	11	2	3	9
14	0	1	4	9	1	6	11	14	0	2	3	13	1	4	9
14	0	1	5	12	2	5	11	14	0	2	4	13	3	9	13
14	0	1	5	12	2	8	13	14	0	2	4	13	7	9	13
14	0	1	6	7	4	6	11	14	0	2	5	6	1	5	11
14	0	1	6	10	3	4	11	14	0	2	5	7	3	10	12
14	0	1	6	13	2	7	8	14	0	2	5	7	4	9	12
14	0	1	8	9	3	4	6	14	0	2	5	11	3	7	12
14	0	1	8	11	5	11	12	14	0	2	5	11	4	8	13
14	0	1	8	11	6	12	13	14	0	2	5	11	4	9	10
14	0	1	10	11	1	9	10	14	0	2	5	11	5	9	10
14	0	2	1	3	5	8	13	14	0	2	5	12	3	7	11
14	0	2	5	13	1	7	12	14	0	3	12	13	4	5	7
14	0	2	7	13	2	4	6	14	0	4	2	3	1	5	11
14	0	2	8	13	1	3	5	14	0	4	3	5	7	10	11
14	0	2	9	11	8	12	13	14	0	4	5	13	1	2	9
14	0	3	1	4	1	4	12	14	0	4	7	9	2	3	6
Continued on next page															

Table 5.17 – continued from previous page

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
14	0	3	1	12	1	3	10	14	0	4	7	9	4	8	11
14	0	3	4	6	4	7	10	14	0	4	9	11	5	10	12
14	0	3	4	8	1	3	13	14	0	5	1	2	2	6	10
14	0	3	4	9	2	4	10	14	0	5	1	4	3	4	11
14	0	3	4	10	5	6	7	14	0	5	2	8	8	10	13
14	0	3	4	10	8	10	11	14	0	5	2	10	7	8	13
14	0	3	4	12	3	12	13	14	0	5	2	11	5	10	11
14	0	3	4	13	8	9	12	14	0	5	6	10	1	5	8
14	0	3	6	7	3	7	10	14	0	5	7	10	2	3	11
14	0	3	6	9	1	6	10	14	0	5	8	10	3	5	8
14	0	3	7	9	1	3	9	14	0	5	8	13	2	12	13
14	0	3	7	9	3	11	13	14	0	5	9	10	1	12	13
14	0	3	10	12	7	9	11	14	0	6	1	5	1	5	8
14	0	6	1	5	10	11	12	14	0	6	3	7	3	6	10
14	0	6	1	9	9	10	11	14	0	6	4	11	9	11	13
14	0	6	1	11	1	6	12	14	0	6	7	12	3	5	11
14	0	6	1	11	2	3	10	14	0	7	2	9	6	7	12
14	0	6	1	12	7	9	11	14	0	7	3	10	3	4	5

Table 5.18 lists the feedback functions of type 5 that have optimal periods for $n = 15$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_3 \cdot x_9 \oplus x_4 \cdot x_6 \cdot x_8$ is a function of type 5 with register size $n = 15$, and is represented in Table 5.18 by the following entry: $[5 \ 0 \ 1 \ 3 \ 9 \ 4 \ 6 \ 8]$. For $n = 15$, we obtained (58) new feedback functions with optimal periods of this type.

Table 5.18: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 15$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
15	0	1	3	9	4	6	8	15	0	2	1	12	3	6	8
15	0	1	3	11	4	10	11	15	0	2	4	10	1	6	7
15	0	1	4	9	9	11	13	15	0	2	8	12	4	9	14
15	0	1	4	11	1	7	14	15	0	3	2	5	7	12	13
15	0	1	5	7	2	4	8	15	0	3	4	7	3	8	9
15	0	1	5	8	7	10	12	15	0	3	7	10	1	6	10
15	0	1	5	9	5	9	12	15	0	3	10	13	9	10	12
15	0	1	5	12	3	6	11	15	0	4	1	3	7	9	11
15	0	1	5	12	5	7	9	15	0	4	2	5	6	12	13
15	0	1	5	12	5	9	12	15	0	4	3	12	3	4	5
15	0	1	5	13	1	8	14	15	0	4	5	6	4	10	11
15	0	1	6	13	3	8	12	15	0	4	5	12	5	6	12
15	0	1	7	12	1	10	12	15	0	4	6	10	8	10	12
15	0	1	8	13	1	5	8	15	0	4	6	12	3	8	11
15	0	1	9	10	5	8	11	15	0	4	11	12	5	10	12
15	0	1	9	13	3	5	7	15	0	4	12	14	3	7	14
15	0	2	1	10	7	12	14	15	0	5	2	7	4	9	13
15	0	5	3	8	6	7	13	15	0	7	3	12	1	9	10
15	0	5	3	13	2	3	11	15	0	7	5	6	2	7	14
15	0	5	3	13	3	4	12	15	0	7	5	7	5	7	14
15	0	5	7	12	5	7	10	15	0	7	5	12	3	9	14
15	0	6	2	5	2	3	14	15	0	7	6	8	5	7	13
15	0	6	2	14	10	12	13	15	0	7	6	11	1	2	3
15	0	6	5	8	7	8	14	15	0	7	6	11	1	2	5
15	0	6	8	11	4	5	10	15	0	7	6	12	4	8	14
15	0	7	3	6	4	10	13	15	0	7	8	11	2	3	11
15	0	7	3	9	8	10	13	15	0	7	9	11	10	12	14
15	0	7	3	10	4	12	13	15	0	7	11	12	8	11	12
15	0	7	3	11	3	8	12	15	0	7	12	13	5	6	14

Table 5.19 lists the feedback functions of type 5 that have optimal periods for $n = 16$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) =$

$x_0 \oplus x_1 \oplus x_2 \cdot x_4 \oplus x_3 \cdot x_4 \cdot x_6$ is a function of type 5 with register size $n = 16$, and

is represented in Table 5.19 by the following entry: [5 0 1 2 4 3 4 6].

A total of (44) new feedback functions for ($n = 16$) have been proposed.

Table 5.19: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 16$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
16	0	1	2	4	3	4	6	16	0	2	13	15	1	2	8
16	0	1	2	8	1	10	15	16	0	3	1	4	1	10	13
16	0	1	4	7	5	7	8	16	0	3	2	6	1	3	14
16	0	1	4	8	1	5	7	16	0	3	2	14	10	11	14
16	0	1	5	8	1	8	12	16	0	3	4	5	10	11	14
16	0	1	7	12	7	8	9	16	0	3	4	15	9	11	12
16	0	1	8	15	4	8	15	16	0	3	6	9	5	12	13
16	0	1	9	14	8	9	14	16	0	3	8	11	4	5	13
16	0	1	10	14	6	7	11	16	0	3	14	15	3	8	14
16	0	2	1	5	5	8	14	16	0	4	7	15	1	4	7
16	0	2	1	11	4	6	15	16	0	4	11	15	6	8	13
16	0	2	1	13	2	10	11	16	0	5	2	11	7	12	15
16	0	2	1	15	4	9	11	16	0	5	4	7	8	11	14
16	0	2	3	15	2	3	9	16	0	5	7	14	7	9	14
16	0	2	5	7	8	9	15	16	0	5	10	14	6	12	14
16	0	2	9	11	7	9	14	16	0	5	12	15	1	5	12
16	0	2	11	15	1	6	15	16	0	6	5	13	7	10	11
16	0	6	9	11	2	3	4	16	0	7	2	15	6	8	11
16	0	7	1	8	2	5	14	16	0	7	4	5	10	11	13
16	0	7	1	10	11	14	15	16	0	7	4	9	3	6	13
16	0	7	1	12	3	4	14	16	0	7	5	12	6	8	11
16	0	7	2	13	3	10	13	16	0	7	9	13	3	11	13

Table 5.20 lists the feedback functions of type 5 that have optimal periods for $n = 17$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_9 \oplus x_3 \cdot x_8 \cdot x_{13}$ is a function of type 5 with register size $n = 17$, and is represented in Table 5.20 by the following entry: $[5 \ 0 \ 1 \ 2 \ 9 \ 3 \ 8 \ 13]$. We obtained (46) new feedback functions with optimal periods of this type for ($n = 17$).

Table 5.20: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 17$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
17	0	1	2	9	3	8	13	17	0	3	1	13	3	12	14
17	0	1	2	12	4	6	12	17	0	3	2	4	11	13	15
17	0	1	3	5	2	4	7	17	0	3	6	15	2	3	11
17	0	1	3	9	5	12	14	17	0	3	7	10	1	3	4
17	0	1	6	15	5	6	8	17	0	3	7	16	1	11	14
17	0	1	6	15	9	10	15	17	0	3	7	16	12	15	16
17	0	1	7	13	4	11	12	17	0	3	13	16	3	7	16
17	0	1	9	10	5	9	14	17	0	4	1	7	8	10	12
17	0	1	11	16	6	7	11	17	0	4	1	12	11	14	15
17	0	2	1	10	1	13	14	17	0	4	2	8	9	11	13
17	0	2	1	10	4	8	9	17	0	4	5	14	1	5	14
17	0	2	4	8	6	10	16	17	0	5	1	14	3	4	7
17	0	2	4	9	6	7	12	17	0	5	2	16	2	4	6
17	0	2	7	14	3	6	10	17	0	5	6	7	7	11	13
17	0	2	8	14	1	12	13	17	0	5	6	11	2	5	8
17	0	2	10	16	4	5	11	17	0	5	11	13	4	5	7
17	0	2	14	16	4	8	12	17	0	6	1	10	2	11	15
17	0	6	4	10	3	13	14	17	0	7	5	12	4	7	11
17	0	6	4	12	4	6	13	17	0	7	11	14	5	7	16
17	0	6	10	16	1	5	15	17	0	8	6	7	3	10	13
17	0	6	12	13	2	13	14	17	0	8	7	14	13	14	15
17	0	7	2	5	8	9	14	17	0	8	11	16	3	6	14
17	0	7	3	9	1	5	16	17	0	8	14	15	6	10	14

Table 5.21 lists the feedback functions of type 5 that have optimal periods for $n = 18$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_6 \cdot x_7 \oplus x_6 \cdot x_{12} \cdot x_{16}$ is a function of type 5 with register size $n = 18$, and is represented in Table 5.21 by the following entry: $[5 \ 0 \ 1 \ 6 \ 7 \ 6 \ 12 \ 16]$. A total of (18) new feedback functions for ($n = 18$) have been yielded.

Table 5.21: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 18$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
18	0	1	6	7	6	12	16	18	0	4	11	13	4	11	17
18	0	1	6	9	5	7	16	18	0	5	7	14	6	7	9
18	0	1	6	10	6	7	16	18	0	5	11	16	3	14	16
18	0	2	1	3	1	8	16	18	0	6	7	13	1	6	7
18	0	2	1	7	5	10	13	18	0	7	2	3	3	8	14
18	0	3	1	10	3	6	13	18	0	7	6	12	13	15	17
18	0	3	5	8	6	11	13	18	0	8	1	12	3	9	17
18	0	4	6	13	1	5	9	18	0	8	3	15	3	16	17
18	0	4	8	11	3	13	15	18	0	8	7	13	4	6	8

Table 5.22 lists the feedback functions of type 5 that have optimal periods for $n = 19$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_2 \cdot x_4 \oplus x_1 \cdot x_3 \cdot x_7$ is a function of type 5 with register size $n = 19$, and

is represented in Table 5.22 by the following entry: $[5 \ 0 \ 1 \ 2 \ 4 \ 1 \ 3 \ 7]$.

For $n = 19$, we obtained (31) new feedback functions with optimal periods of this type.

Table 5.22: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 19$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
19	0	1	2	4	1	3	7	19	0	3	9	14	2	9	10
19	0	1	7	12	2	11	13	19	0	4	1	2	2	5	8
19	0	1	7	17	2	5	12	19	0	4	7	16	5	12	18
19	0	1	9	11	12	14	17	19	0	4	9	10	6	10	14
19	0	1	10	11	6	11	13	19	0	4	14	16	2	10	15
19	0	1	10	15	6	8	13	19	0	5	1	6	5	7	16
19	0	1	10	15	8	15	16	19	0	5	1	11	1	6	16
19	0	1	12	14	2	8	12	19	0	5	3	13	3	5	8
19	0	1	13	16	1	4	15	19	0	5	8	11	3	14	15
19	0	2	1	4	10	11	14	19	0	5	9	11	15	16	18
19	0	2	8	9	10	12	16	19	0	6	9	10	4	5	6
19	0	2	8	14	2	9	18	19	0	6	10	15	3	8	16
19	0	2	12	18	5	11	16	19	0	7	2	13	3	8	14
19	0	2	14	16	5	11	14	19	0	7	10	11	1	6	18
19	0	3	5	6	4	9	17	19	0	9	3	11	2	10	14
19	0	3	5	14	2	8	17								

Table 5.23 lists the feedback functions of type 5 that have optimal periods for $n = 20$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_7 \cdot x_{10} \oplus x_3 \cdot x_{12} \cdot x_{14}$ is a function of type 5 with register size $n = 20$, and is represented in Table 5.23 by the following entry: $[5 \ 0 \ 1 \ 7 \ 10 \ 3 \ 12 \ 14]$. A total of (13) new feedback functions for ($n = 20$) have been constructed.

Table 5.23: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 20$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
20	0	1	7	10	3	12	14	20	0	7	3	18	1	18	19
20	0	1	10	11	5	10	12	20	0	7	8	11	13	15	16
20	0	2	5	9	4	8	17	20	0	7	11	18	10	11	14
20	0	2	7	9	2	14	17	20	0	8	5	17	1	6	13
20	0	3	4	12	5	12	17	20	0	9	3	18	10	14	16
20	0	3	10	15	2	8	12	20	0	9	11	13	3	9	15
20	0	5	2	12	2	3	10								

Table 5.24 lists the feedback functions of type 5 that have optimal periods for $n = 21$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_1 \oplus x_{10} \cdot x_{14} \oplus x_3 \cdot x_7 \cdot x_{11}$ is a function of type 5 with register size $n = 21$, and is represented in Table 5.24 by the following entry: $[5 \ 0 \ 1 \ 10 \ 14 \ 3 \ 7 \ 11]$. We obtained (13) new feedback functions with optimal periods of this type for ($n = 21$).

Table 5.24: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 21$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
21	0	1	10	14	3	7	11	21	0	7	3	17	2	4	5
21	0	2	3	14	9	18	19	21	0	8	1	10	6	10	18
21	0	2	10	18	6	8	19	21	0	8	3	6	10	18	20
21	0	3	5	13	4	7	16	21	0	8	4	7	11	12	13
21	0	3	5	20	3	8	14	21	0	9	1	19	4	8	14
21	0	5	10	15	2	17	18	21	0	10	3	8	9	16	18
21	0	5	16	19	1	4	7								

Table 5.25 lists the feedback functions of type 5 that have optimal periods for $n = 22$. Each function in this table is of the form: $[n \ 0 \ a \ b \ c \ d \ e \ h]$, where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. For example: $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = x_0 \oplus x_2 \oplus x_5 \cdot x_7 \oplus x_1 \cdot x_{13} \cdot x_{18}$ is a function of type 5 with register size $n = 22$, and is represented in Table 5.25 by the following entry: $[5 \ 0 \ 2 \ 5 \ 7 \ 1 \ 13 \ 18]$. A total of (8) new feedback functions for ($n = 22$) have been proposed.

Table 5.25: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 22$

n	0	a	b	c	d	e	h	n	0	a	b	c	d	e	h
22	0	2	5	7	1	13	18	22	0	6	3	15	2	20	21
22	0	5	1	4	2	9	14	22	0	6	3	15	12	16	17
22	0	5	11	20	12	14	20	22	0	6	3	21	3	6	16
22	0	5	18	21	14	18	20	22	0	7	4	21	6	13	19

Tables [5.26, 5.27, 5.28] list the feedback functions of type 5 that have optimal periods for $n = 23, 24$ and 25 respectively. Each entry in these tables is of the form: $[n \ type \ 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (5.4) in section 5.2. This type of feedback function is of algebraic degree 3 because of the number of variables in the largest product term (last term) of the presented functions in these tables is equal three. We obtained (3, 2 and 3) new feedback functions with optimal periods of this type for $n = 23, 24$ and 25 respectively, which are

constructed by our construction method (*SFG*) described in section 4.3.

Table 5.26: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 23$

n	$type$	0	a	b	c	d	e	h
23	5	0	7	5	9	8	12	22
23	5	0	8	14	22	3	7	16
23	5	0	9	6	16	2	6	16

Table 5.27: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 24$

n	$type$	0	a	b	c	d	e	h
24	5	0	1	3	6	2	10	14
24	5	0	2	19	21	12	13	14

Table 5.28: *NLFSR* Functions (Type 5) with Optimal Periods, $n = 25$

n	$type$	0	a	b	c	d	e	h
25	5	0	6	10	14	9	13	14
25	5	0	7	5	20	5	8	12
25	5	0	11	5	6	9	13	24

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis proposes new *NLFSR* functions with optimal periods for different five types of feedback functions. The methodology used in this thesis follows four phases. In phase 1, we studied the importance of *NLFSRs* and their security strength in recent applications such as stream cipher cryptosystems and pseudo-random number generators of many cryptographic algorithms. In phase 2, we implemented *NLFSR* period verifier system to test a list of feedback functions and to find their periods. Then, we verified the existing feedback functions of various sizes, n that provide optimal periods for *NLFSRs* in phase 3. This phase also helps verifying the correctness of the implementation in phase 2. Finally, in phase 4, we constructed new *NLFSR* functions with optimal periods for the existing types of feedback functions of algebraic degree 2 and 3, which have simple algebraic normal form. Moreover, we proposed new type of feedback functions of

algebraic degree 3 having optimal periods for $n \leq 25$.

In this chapter, we summarize the work of this thesis in section 6.1. Moreover, we discuss the important results and their indications in section 6.2. In the end, section 6.3 introduces some outlines for the future work.

6.1 Thesis Summary

Nonlinear feedback shift registers (*NLFSRs*) have received much attention in designing various cryptographic algorithms such as stream ciphers and light weight block ciphers to provide high-level security in communication systems. In this work, the importance of *NLFSRs* and their security strength in recent applications have been studied and compared to the *LFSRs*. An *NLFSR* period verifier program is implemented to verify the existing *NLFSR* feedback functions. The previous study done by Dubrova has been verified which provides a list of *NLFSR* feedback functions in four predefined types with optimal period of $2^n - 1$, for $(4 \leq n \leq 24)$, where n is the size of the register. This study is considered the only and best so far study that mainly focused on exploring *NLFSR* feedback functions with optimal periods. This study constructed *NLFSR* functions of different three types of feedback functions. Therefore, we choose this study as one of the main references in our work, especially for the verification phase.

As a result of our verification phase, we found that Dubrova's work did not list all the feedback functions having optimal periods for many values of n in the specified range. Therefore, this work closes the gap by constructing the missing

feedback functions with optimal periods for all $n = 4, 5, \dots, 25$. Moreover, a new type of feedback function (type 5) is proposed, that also provides optimal periods for *NLFSRs* for different values of $n \leq 25$.

The construction method of these functions is briefly explained. This construction method consists of two steps: the enumeration step and the period-testing step. The enumeration step uses Sequential Function Generator (*SFG*) method which enumerates feedback functions by incrementing the subscripts of a given feedback function. Then the period-testing algorithm selects a start point of the states of the *NLFSRs* and determines the period of the sequence by state-transforming operations until the start point appears again. Finally, we keep all the n -stage feedback functions generating sequences whose periods achieving $2^n - 1$.

The contributions of this thesis are indicated by dotted lines in Figure 6.1. *NLFSR* feedback functions of all the five types considered in this work yield (1718) new feedback functions with optimal periods. For type 1, 59 new feedback functions are proposed. In addition, 212, 169 and 140 new feedback functions of types 2, 3 and 4 are constructed, respectively. Moreover, for type 5, 1138 new feedback functions are constructed.

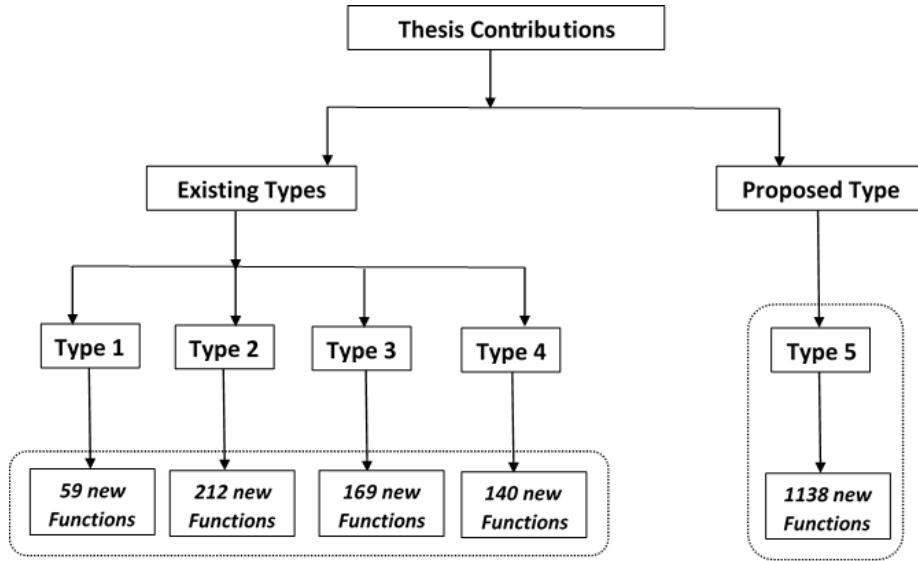


Figure 6.1: Taxonomy of Thesis Contributions.

6.2 Concluding Remarks

In this thesis, (1718) new *NLFSR* feedback functions with optimal periods were proposed for all the five types considered in our work. All these functions are either of algebraic degree 2 or 3, which have a fairly simple algebraic normal form. Therefore, these functions are desirable for many applications since the circuit complexity of feedback functions are not increased.

Modern stream ciphers can make use of one *NLFSR* feedback function as the case of Grain stream cipher, or many *NLFSR* feedback functions such as Trivium stream cipher to increase the complexity of the output sequences generated by the periodic state of *NLFSR*, and therefore, increasing the complexity of the key stream in stream cipher cryptosystems.

The importance of this work lies on generating new feedback functions that can significantly improve the cryptographic strength of the generators of stream ci-

pher cryptosystems and pseudorandom number generators of many cryptographic algorithms. Moreover, numerous network security applications based on cryptographic algorithms can make use of random numbers for different objectives, such as generation of keys for symmetric key cryptography, hash function algorithms, public key cryptography, digital signature algorithms and network security protocols.

6.3 Future Work

In this section, we suggest extending this work in the following ways. One suggested approach to extend this work is to explore different types of feedback functions with simple algebraic normal form to be desirable for implementation. Providing different types of feedback function will give the designers the variety and the freedom to choose the appropriate *NLFSR* feedback functions based on the environment of cryptographic applications that *NLFSR* based stream cipher will be used in.

Another way to extend this work is by constructing feedback function with larger values of n for new types. Therefore, strengthening the complexity of the generators of stream cipher cryptosystems. In addition, the searching method can be improved by using Field Programmable Gate Arrays (*FPGAs*) to reduce the search time of *NLFSRs* with optimal period. Using *FPGAs* will improve and optimize the performance as compared with using only the personal computer *CPU* resources.

CHAPTER 7

APPENDIX

7.1 List of Proposed *NLFSR* Feedback Functions with Optimal Periods

In this section, Table 7.1 lists all the new feedback functions with optimal periods for the functions of types 1, 2, 3 and 4 in the range ($4 \leq n \leq 25$). Any missing entry in this table implies that no new feedback functions with optimal periods were found for that particular type and value of n .

Each function in this table is of the form: $[n \text{ type } 0 \ a \ b \ c \ d \ e \ h]$ where n is the size of the *NLFSR*, and a, b, c, d, e and h are the indices of the variables, as defined in (4.1), (4.2), (4.3) and (4.4) in section 4.3. Notice that e and h are ignored in type 1, while h is ignored in types 2 and 4, and indicated by N/A (Not Applicable).

Table 7.1: *NLFSR* Functions with Optimal Periods for ($4 \leq n \leq 25$)

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
4	1	0	2	3	1	2	N/A	N/A	4	2	0	2	1	3	2	3	N/A
4	1	0	2	3	1	3	N/A	N/A	4	2	0	3	1	2	2	3	N/A
4	1	0	2	3	2	3	N/A	N/A									
5	1	0	2	4	1	4	N/A	N/A	5	2	0	3	1	4	2	3	N/A
5	1	0	2	4	2	3	N/A	N/A	5	2	0	4	1	2	3	4	N/A
5	1	0	2	4	2	4	N/A	N/A	5	2	0	4	1	3	2	4	N/A
5	1	0	3	4	1	3	N/A	N/A	5	2	0	4	1	4	2	3	N/A
5	2	0	3	1	2	3	4	N/A	5	2	0	4	2	3	2	4	N/A
5	2	0	3	1	3	1	4	N/A	5	2	0	4	2	3	3	4	N/A
5	2	0	3	1	3	2	4	N/A	5	4	0	2	4	1	2	3	N/A
5	4	0	2	4	1	2	4	N/A	5	4	0	2	4	2	3	4	N/A
5	4	0	2	4	1	3	4	N/A									
6	1	0	2	5	2	5	N/A	N/A	6	2	0	4	1	2	1	5	N/A
6	1	0	3	4	1	5	N/A	N/A	6	2	0	4	1	3	2	3	N/A
6	1	0	3	4	2	4	N/A	N/A	6	2	0	4	1	3	3	4	N/A
Continued on next page																	

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
6	1	0	3	4	3	4	N/A	N/A	6	2	0	4	1	3	3	5	N/A
6	1	0	3	4	3	5	N/A	N/A	6	2	0	4	1	5	2	4	N/A
6	1	0	3	5	1	5	N/A	N/A	6	2	0	4	2	3	3	5	N/A
6	1	0	4	5	2	4	N/A	N/A	6	2	0	4	2	4	3	5	N/A
6	1	0	4	5	4	5	N/A	N/A	6	2	0	5	1	2	4	5	N/A
6	2	0	3	2	3	2	5	N/A	6	2	0	5	1	3	3	5	N/A
6	2	0	3	2	4	2	5	N/A	6	2	0	5	1	4	3	4	N/A
6	2	0	3	2	5	3	4	N/A	6	3	0	1	2	3	4	1	5
6	3	0	1	2	3	4	3	5	6	4	0	3	4	2	3	4	N/A
6	3	0	1	2	3	5	1	4	6	4	0	3	4	2	3	5	N/A
6	3	0	1	2	3	5	2	3	6	4	0	3	4	2	4	5	N/A
6	3	0	1	3	4	5	2	5	6	4	0	4	5	1	2	4	N/A
6	3	0	1	3	4	5	3	4	6	4	0	4	5	1	2	5	N/A
6	3	0	2	3	4	5	1	3	6	4	0	4	5	2	3	4	N/A
6	3	0	2	3	4	5	1	5	6	4	0	4	5	2	4	5	N/A
6	4	0	3	4	1	3	5	N/A									
7	1	0	2	6	1	3	N/A	N/A	7	1	0	3	6	4	6	N/A	N/A
7	1	0	2	6	2	4	N/A	N/A	7	1	0	5	6	1	5	N/A	N/A
7	1	0	2	6	2	6	N/A	N/A	7	2	0	4	1	4	3	6	N/A
7	1	0	3	5	2	5	N/A	N/A	7	2	0	4	1	6	3	4	N/A
7	1	0	3	5	5	6	N/A	N/A	7	2	0	4	1	6	4	6	N/A
7	2	0	4	2	3	4	5	N/A	7	2	0	6	2	4	2	5	N/A
7	2	0	4	2	4	2	5	N/A	7	2	0	6	2	4	4	5	N/A
7	2	0	4	2	4	2	6	N/A	7	2	0	6	2	6	3	4	N/A
7	2	0	4	4	5	5	6	N/A	7	3	0	1	4	5	6	1	5
7	2	0	5	1	2	1	4	N/A	7	3	0	1	4	5	6	2	6
7	2	0	5	1	2	1	6	N/A	7	3	0	1	4	5	6	4	6
7	2	0	5	1	3	1	5	N/A	7	3	0	2	3	5	6	1	5
7	2	0	5	1	3	2	5	N/A	7	3	0	2	3	5	6	2	6
7	2	0	5	1	3	5	6	N/A	7	3	0	2	3	5	6	5	6
7	2	0	5	1	4	1	6	N/A	7	3	0	3	4	5	6	1	5
7	2	0	5	1	5	2	6	N/A	7	3	0	3	4	5	6	1	6
7	2	0	5	1	6	3	5	N/A	7	3	0	3	4	5	6	4	5
7	2	0	5	2	4	3	5	N/A	7	4	0	2	6	1	3	5	N/A
7	2	0	5	3	4	3	6	N/A	7	4	0	2	6	1	3	6	N/A
7	2	0	6	1	2	5	6	N/A	7	4	0	2	6	1	4	5	N/A
7	2	0	6	1	6	2	3	N/A	7	4	0	3	5	1	2	3	N/A
7	2	0	6	2	3	2	5	N/A	7	4	0	3	5	1	2	5	N/A
7	2	0	6	2	3	3	4	N/A	7	4	0	3	5	1	3	4	N/A

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
7	4	0	3	5	1	4	6	N/A	7	4	0	3	5	3	5	6	N/A
7	4	0	3	5	2	4	5	N/A	7	4	0	3	6	2	4	6	N/A
7	4	0	3	5	2	4	6	N/A									
8	1	0	2	7	1	7	N/A	N/A	8	1	0	4	5	1	4	N/A	N/A
8	1	0	2	7	2	3	N/A	N/A	8	1	0	4	5	1	6	N/A	N/A
8	1	0	2	7	3	4	N/A	N/A	8	1	0	4	5	2	4	N/A	N/A
8	1	0	2	7	4	6	N/A	N/A	8	1	0	4	5	4	5	N/A	N/A
8	1	0	2	7	6	7	N/A	N/A	8	1	0	4	5	4	6	N/A	N/A
8	1	0	3	6	1	5	N/A	N/A	8	2	0	5	1	2	4	6	N/A
8	1	0	3	6	3	4	N/A	N/A	8	2	0	5	1	4	2	7	N/A
8	1	0	3	6	4	6	N/A	N/A	8	2	0	5	1	5	2	6	N/A
8	1	0	3	7	3	7	N/A	N/A	8	2	0	5	2	3	5	6	N/A
8	1	0	4	5	1	2	N/A	N/A	8	2	0	5	2	4	2	7	N/A
8	2	0	5	2	5	2	7	N/A	8	3	0	2	3	4	5	2	6
8	2	0	5	4	6	4	7	N/A	8	3	0	2	3	4	6	4	5
8	2	0	5	4	6	6	7	N/A	8	3	0	2	4	5	6	3	4
8	2	0	6	1	2	1	5	N/A	8	3	0	2	5	6	7	1	3
8	2	0	6	1	2	3	7	N/A	8	3	0	2	5	6	7	3	5
8	2	0	6	1	3	5	7	N/A	8	3	0	3	4	5	6	2	6
8	2	0	6	1	7	5	6	N/A	8	3	0	3	4	6	7	4	6
8	2	0	6	2	4	5	7	N/A	8	3	0	3	5	6	7	2	6
8	2	0	7	1	2	4	6	N/A	8	4	0	2	7	1	2	7	N/A
8	2	0	7	1	4	4	5	N/A	8	4	0	2	7	2	3	5	N/A
8	2	0	7	2	7	3	6	N/A	8	4	0	2	7	2	3	7	N/A
8	2	0	7	4	6	4	7	N/A	8	4	0	2	7	2	4	5	N/A
8	2	0	7	4	6	5	6	N/A	8	4	0	2	7	2	5	6	N/A
8	3	0	1	3	6	7	4	6	8	4	0	2	7	2	6	7	N/A
8	3	0	1	4	5	7	1	5	8	4	0	2	7	3	4	7	N/A
8	3	0	1	4	5	7	2	7	8	4	0	4	5	2	5	7	N/A
8	3	0	1	4	5	7	4	7	8	4	0	4	5	3	4	7	N/A
8	3	0	1	4	6	7	3	7	8	4	0	4	5	4	5	7	N/A
8	4	0	4	7	1	3	7	N/A	8	4	0	6	7	1	2	3	N/A
8	4	0	4	7	1	4	5	N/A	8	4	0	6	7	1	5	6	N/A
8	4	0	4	7	1	6	7	N/A	8	4	0	6	7	2	3	4	N/A
8	4	0	4	7	2	4	6	N/A	8	4	0	6	7	2	3	6	N/A
8	4	0	4	7	3	4	5	N/A	8	4	0	6	7	3	6	7	N/A
8	4	0	4	7	4	6	7	N/A	8	4	0	6	7	4	5	6	N/A
9	1	0	3	8	1	5	N/A	N/A	9	2	0	5	1	7	4	7	N/A
9	1	0	3	8	3	5	N/A	N/A	9	2	0	5	1	7	6	8	N/A

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
9	1	0	5	6	2	6	N/A	N/A	9	2	0	5	2	6	4	6	N/A
9	1	0	5	7	4	5	N/A	N/A	9	2	0	5	3	6	3	8	N/A
9	2	0	5	1	4	6	7	N/A	9	2	0	6	1	5	2	8	N/A
9	2	0	5	1	6	2	7	N/A	9	2	0	6	2	5	6	7	N/A
9	2	0	5	1	7	2	3	N/A	9	2	0	6	2	5	7	8	N/A
9	2	0	6	2	8	3	8	N/A	9	3	0	2	4	6	8	3	4
9	2	0	7	1	8	5	6	N/A	9	3	0	2	5	6	7	1	7
9	2	0	7	2	5	3	4	N/A	9	3	0	2	5	7	8	3	8
9	2	0	7	2	7	3	5	N/A	9	3	0	2	6	7	8	3	5
9	2	0	7	2	7	3	8	N/A	9	3	0	3	4	7	8	3	7
9	2	0	7	3	5	4	8	N/A	9	3	0	3	4	7	8	3	8
9	2	0	8	1	4	2	6	N/A	9	3	0	4	5	6	8	2	6
9	2	0	8	1	6	6	7	N/A	9	3	0	5	6	7	8	2	6
9	2	0	8	1	7	2	6	N/A	9	4	0	3	7	2	3	8	N/A
9	2	0	8	1	8	2	7	N/A	9	4	0	3	7	3	7	8	N/A
9	2	0	8	1	8	3	4	N/A	9	4	0	3	7	4	5	6	N/A
9	2	0	8	2	3	3	8	N/A	9	4	0	4	6	1	2	5	N/A
9	2	0	8	4	6	5	6	N/A	9	4	0	4	6	1	7	8	N/A
9	2	0	8	4	7	4	8	N/A	9	4	0	4	6	2	5	6	N/A
9	3	0	1	4	6	8	4	6	9	4	0	4	6	3	4	5	N/A
9	3	0	1	4	7	8	3	7	9	4	0	4	6	4	5	6	N/A
9	3	0	2	3	5	8	2	8	9	4	0	4	6	4	5	8	N/A
9	3	0	2	3	7	8	3	6	9	4	0	5	7	1	3	4	N/A
9	4	0	5	7	4	7	8	N/A	9	4	0	6	8	1	3	5	N/A
9	4	0	6	8	1	2	8	N/A	9	4	0	6	8	3	6	7	N/A
10	1	0	2	9	3	4	N/A	N/A	10	2	0	6	2	3	7	9	N/A
10	1	0	5	6	1	6	N/A	N/A	10	2	0	6	3	9	7	9	N/A
10	1	0	5	6	2	6	N/A	N/A	10	2	0	7	1	2	3	4	N/A
10	1	0	5	6	4	8	N/A	N/A	10	2	0	7	1	2	3	8	N/A
10	1	0	5	8	5	9	N/A	N/A	10	2	0	7	1	3	2	8	N/A
10	1	0	6	9	3	7	N/A	N/A	10	2	0	7	1	3	7	9	N/A
10	1	0	8	9	1	2	N/A	N/A	10	2	0	7	1	4	4	9	N/A
10	2	0	6	1	3	2	3	N/A	10	2	0	7	2	7	4	9	N/A
10	2	0	6	1	3	5	9	N/A	10	2	0	7	2	8	8	9	N/A
10	2	0	6	1	3	7	9	N/A	10	2	0	7	4	8	7	8	N/A
10	2	0	6	1	9	5	9	N/A	10	2	0	8	1	2	1	7	N/A
10	2	0	8	1	5	1	9	N/A	10	3	0	2	6	8	9	5	9
10	2	0	8	1	6	5	9	N/A	10	3	0	2	6	8	9	6	8
10	2	0	8	3	6	7	9	N/A	10	3	0	3	4	5	6	2	6

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
10	2	0	8	3	7	5	7	N/A	10	3	0	3	4	5	7	2	6
10	2	0	8	3	7	6	9	N/A	10	3	0	3	4	5	7	4	6
10	2	0	8	3	9	4	6	N/A	10	3	0	3	4	5	9	2	8
10	2	0	8	3	9	4	9	N/A	10	3	0	3	4	6	8	4	9
10	2	0	8	5	7	5	9	N/A	10	3	0	3	4	7	9	4	9
10	2	0	9	1	3	2	8	N/A	10	3	0	3	5	6	7	4	6
10	2	0	9	2	6	3	4	N/A	10	3	0	3	5	6	7	4	8
10	2	0	9	2	7	3	6	N/A	10	3	0	3	6	7	9	4	7
10	2	0	9	5	8	6	8	N/A	10	3	0	4	5	6	7	4	8
10	2	0	9	6	7	8	9	N/A	10	3	0	4	6	7	8	4	7
10	3	0	1	5	6	9	1	5	10	4	0	2	9	1	2	5	N/A
10	3	0	1	5	6	9	1	6	10	4	0	4	8	1	5	9	N/A
10	3	0	1	5	6	9	1	9	10	4	0	5	8	1	3	4	N/A
10	3	0	2	5	6	8	6	8	10	4	0	5	8	3	6	8	N/A
10	3	0	2	5	8	9	1	5	10	4	0	5	8	6	8	9	N/A
10	4	0	5	9	2	4	6	N/A	10	4	0	7	8	2	5	7	N/A
10	4	0	6	9	1	2	6	N/A	10	4	0	8	9	1	4	9	N/A
10	4	0	6	9	1	5	7	N/A	10	4	0	8	9	4	5	9	N/A
10	4	0	6	9	4	8	9	N/A	10	4	0	8	9	6	8	9	N/A
11	1	0	2	10	7	10	N/A	N/A	11	2	0	8	1	4	4	8	N/A
11	1	0	3	9	2	5	N/A	N/A	11	2	0	8	1	6	2	10	N/A
11	1	0	6	9	2	10	N/A	N/A	11	2	0	8	2	3	5	10	N/A
11	2	0	6	2	5	7	10	N/A	11	2	0	8	2	5	6	8	N/A
11	2	0	6	3	5	3	9	N/A	11	2	0	8	3	6	5	8	N/A
11	2	0	6	4	5	4	7	N/A	11	2	0	8	4	6	4	9	N/A
11	2	0	7	1	2	9	10	N/A	11	2	0	9	1	2	5	7	N/A
11	2	0	7	1	9	8	9	N/A	11	2	0	9	1	3	2	4	N/A
11	2	0	7	3	7	4	8	N/A	11	2	0	9	1	5	6	7	N/A
11	2	0	10	1	8	4	9	N/A	11	3	0	3	4	8	9	1	7
11	2	0	10	2	10	4	9	N/A	11	3	0	3	4	9	10	1	2
11	2	0	10	3	4	4	8	N/A	11	3	0	3	4	9	10	2	10
11	2	0	10	3	8	4	8	N/A	11	3	0	3	5	6	10	2	6
11	2	0	10	3	9	4	10	N/A	11	3	0	3	5	8	10	3	5
11	2	0	10	6	7	8	9	N/A	11	3	0	3	6	7	10	4	6
11	2	0	10	6	9	7	8	N/A	11	3	0	4	5	6	8	3	7
11	3	0	1	4	7	10	2	10	11	3	0	4	5	7	8	8	9
11	3	0	1	5	8	10	2	4	11	3	0	4	7	9	10	2	3
11	3	0	1	7	8	10	1	5	11	3	0	4	7	9	10	2	7
11	3	0	1	7	9	10	2	8	11	3	0	4	7	9	10	8	9

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
11	3	0	1	7	9	10	2	10	11	3	0	5	6	7	8	1	9
11	3	0	2	3	5	10	5	9	11	3	0	6	7	9	10	5	7
11	3	0	2	4	6	10	3	9	11	3	0	6	8	9	10	5	7
11	3	0	2	4	8	10	3	10	11	4	0	2	10	3	8	10	N/A
11	3	0	2	6	7	9	2	6	11	4	0	4	8	1	7	10	N/A
11	3	0	3	4	8	9	1	4	11	4	0	4	10	1	2	10	N/A
11	3	0	3	4	8	9	1	5	11	4	0	4	10	3	4	7	N/A
11	4	0	4	10	3	5	6	N/A	11	4	0	7	9	2	3	9	N/A
11	4	0	4	10	4	6	9	N/A	11	4	0	7	9	4	9	10	N/A
11	4	0	5	7	1	3	4	N/A	11	4	0	7	10	2	6	10	N/A
11	4	0	5	7	4	7	9	N/A	11	4	0	8	10	2	8	9	N/A
11	4	0	7	9	1	3	8	N/A	11	4	0	8	10	6	7	10	N/A
11	4	0	7	9	2	3	4	N/A									
12	1	0	4	9	3	9	N/A	N/A	12	2	0	10	1	9	3	9	N/A
12	1	0	5	8	5	8	N/A	N/A	12	2	0	10	1	11	2	11	N/A
12	1	0	5	8	5	11	N/A	N/A	12	2	0	10	3	5	9	10	N/A
12	2	0	9	1	7	2	6	N/A	12	2	0	10	3	7	3	9	N/A
12	2	0	9	2	5	4	11	N/A	12	2	0	10	4	10	5	11	N/A
12	2	0	10	1	4	1	7	N/A	12	2	0	10	6	9	9	11	N/A
12	2	0	10	1	5	3	5	N/A	12	2	0	11	1	4	2	3	N/A
12	2	0	11	2	4	4	5	N/A	12	3	0	4	6	7	11	6	8
12	2	0	11	2	6	4	10	N/A	12	3	0	4	6	9	10	6	9
12	2	0	11	2	9	7	10	N/A	12	3	0	5	6	9	11	2	8
12	2	0	11	8	9	9	10	N/A	12	3	0	7	9	10	11	3	7
12	3	0	1	6	7	11	4	5	12	4	0	3	10	1	5	6	N/A
12	3	0	1	6	10	11	6	10	12	4	0	3	10	2	3	5	N/A
12	3	0	2	3	7	11	5	6	12	4	0	3	10	4	7	11	N/A
12	3	0	2	4	8	11	7	10	12	4	0	4	9	1	4	11	N/A
12	3	0	2	5	9	11	7	8	12	4	0	4	11	2	4	11	N/A
12	3	0	2	6	7	10	2	10	12	4	0	6	7	4	7	10	N/A
12	3	0	2	6	9	10	2	8	12	4	0	6	11	1	2	5	N/A
12	3	0	2	6	9	10	6	10	12	4	0	6	11	5	7	10	N/A
12	3	0	2	8	9	10	4	9	12	4	0	6	11	6	7	10	N/A
12	3	0	3	5	7	11	1	11	12	4	0	6	11	6	7	11	N/A
12	3	0	3	6	9	11	2	8	12	4	0	7	8	1	2	8	N/A
12	3	0	3	6	9	11	3	11	12	4	0	7	8	2	3	4	N/A
12	3	0	3	7	10	11	1	5	12	4	0	7	8	5	8	11	N/A
12	3	0	4	6	7	11	2	6	12	4	0	8	9	4	5	11	N/A
12	4	0	9	10	3	7	11	N/A	12	4	0	10	11	2	3	10	N/A

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
12	4	0	9	11	3	5	11	N/A									
13	1	0	2	12	4	8	N/A	N/A	13	2	0	10	2	4	7	9	N/A
13	1	0	5	9	3	4	N/A	N/A	13	2	0	10	2	11	4	12	N/A
13	2	0	7	1	11	8	12	N/A	13	2	0	10	3	4	4	5	N/A
13	2	0	8	1	10	2	4	N/A	13	2	0	11	1	3	3	6	N/A
13	2	0	8	1	12	2	6	N/A	13	2	0	11	1	4	7	12	N/A
13	2	0	8	4	9	7	10	N/A	13	2	0	12	2	7	10	11	N/A
13	2	0	8	4	9	8	12	N/A	13	2	0	12	2	8	8	11	N/A
13	2	0	8	4	11	8	9	N/A	13	2	0	12	4	11	8	9	N/A
13	2	0	9	1	3	10	12	N/A	13	2	0	12	5	7	7	11	N/A
13	2	0	9	3	5	4	11	N/A	13	2	0	12	6	10	6	12	N/A
13	2	0	9	7	9	10	12	N/A	13	3	0	2	3	6	12	7	11
13	3	0	2	3	11	12	1	7	13	3	0	7	9	10	12	3	7
13	3	0	2	5	10	11	3	12	13	3	0	8	9	11	12	6	12
13	3	0	2	6	8	12	3	5	13	4	0	5	11	1	3	5	N/A
13	3	0	2	7	8	11	2	5	13	4	0	6	10	1	4	7	N/A
13	3	0	3	6	7	11	1	5	13	4	0	6	12	2	8	12	N/A
13	3	0	3	6	8	10	3	11	13	4	0	6	12	3	4	10	N/A
13	3	0	3	7	8	10	2	5	13	4	0	6	12	7	8	9	N/A
13	3	0	3	8	9	12	5	9	13	4	0	8	12	1	2	10	N/A
13	3	0	4	6	8	12	4	5	13	4	0	11	12	3	5	7	N/A
13	3	0	6	7	8	12	4	8									
14	1	0	12	13	2	7	N/A	N/A	14	2	0	9	1	8	10	12	N/A
14	2	0	8	1	2	5	9	N/A	14	2	0	11	1	8	2	12	N/A
14	2	0	8	1	13	5	9	N/A	14	2	0	11	2	7	4	9	N/A
14	2	0	11	2	8	10	12	N/A	14	3	0	6	9	10	13	6	12
14	2	0	11	2	10	8	13	N/A	14	3	0	7	9	11	12	9	13
14	2	0	12	3	11	9	13	N/A	14	3	0	7	10	12	13	11	13
14	2	0	13	1	12	2	10	N/A	14	3	0	9	11	12	13	11	13
14	2	0	13	2	5	2	9	N/A	14	4	0	2	13	3	7	9	N/A
14	3	0	1	9	10	13	8	13	14	4	0	4	13	10	11	13	N/A
14	3	0	2	4	8	13	5	7	14	4	0	5	10	9	11	12	N/A
14	3	0	2	4	8	13	7	11	14	4	0	5	10	9	12	13	N/A
14	3	0	2	4	11	12	4	5	14	4	0	6	9	2	3	13	N/A
14	3	0	2	5	7	13	1	11	14	4	0	6	9	5	6	13	N/A
14	3	0	2	8	9	12	4	8	14	4	0	7	10	2	6	12	N/A
14	3	0	3	5	7	12	2	3	14	4	0	7	10	4	6	10	N/A
14	3	0	3	7	10	13	3	13	14	4	0	8	11	5	6	10	N/A
14	3	0	4	7	8	10	1	9	14	4	0	8	11	6	12	13	N/A

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
14	3	0	6	8	9	10	10	13	14	4	0	12	13	6	10	12	N/A
15	1	0	6	10	4	13	N/A	N/A	15	3	0	2	6	12	13	8	12
15	2	0	8	2	5	4	13	N/A	15	3	0	3	10	11	14	11	12
15	2	0	8	2	12	3	12	N/A	15	3	0	5	8	10	11	1	14
15	2	0	9	2	3	4	7	N/A	15	3	0	5	10	11	12	8	12
15	2	0	11	1	3	7	8	N/A	15	3	0	7	8	10	12	2	12
15	2	0	11	1	10	9	10	N/A	15	4	0	5	13	8	9	10	N/A
15	2	0	11	4	13	5	8	N/A	15	4	0	7	9	2	3	12	N/A
15	2	0	11	5	6	5	9	N/A	15	4	0	7	11	2	6	7	N/A
15	2	0	13	1	3	7	9	N/A	15	4	0	12	14	2	3	8	N/A
16	1	0	3	14	13	14	N/A	N/A	16	2	0	11	4	10	8	12	N/A
16	2	0	9	2	8	4	5	N/A	16	2	0	11	4	12	8	9	N/A
16	2	0	9	3	6	10	14	N/A	16	2	0	13	2	9	3	14	N/A
16	2	0	13	9	11	11	15	N/A	16	3	0	6	7	8	12	4	8
16	3	0	2	3	11	15	1	2	16	3	0	6	8	9	14	10	13
16	3	0	2	6	11	14	2	10	16	3	0	7	13	14	15	2	10
16	3	0	3	4	5	15	1	11	16	4	0	8	9	2	4	11	N/A
16	3	0	3	8	9	14	1	13	16	4	0	8	9	6	8	10	N/A
16	3	0	4	5	10	14	1	2	16	4	0	8	15	5	8	9	N/A
17	2	0	9	1	6	5	7	N/A	17	3	0	3	5	14	16	7	15
17	2	0	10	3	8	9	16	N/A	17	3	0	5	8	13	15	1	11
17	2	0	11	5	10	8	15	N/A	17	3	0	5	8	14	16	4	10
17	2	0	12	4	11	10	13	N/A	17	3	0	6	8	12	16	4	16
17	2	0	14	3	4	8	11	N/A	17	3	0	7	10	11	14	2	8
17	2	0	16	2	8	7	10	N/A	17	4	0	10	13	2	14	16	N/A
18	2	0	15	7	16	11	12	N/A	18	3	0	7	9	10	17	3	16
18	2	0	17	6	7	6	9	N/A	18	3	0	11	13	14	17	8	16
18	3	0	3	6	12	16	3	7	18	4	0	9	16	1	12	13	N/A
18	3	0	5	7	9	15	2	14	18	4	0	10	11	6	8	9	N/A
18	3	0	6	7	13	17	9	17	18	4	0	10	17	3	6	8	N/A
19	2	0	13	1	2	11	15	N/A	19	3	0	5	7	13	14	1	17
19	2	0	15	3	14	5	12	N/A	19	3	0	11	14	15	18	4	14
19	2	0	17	9	11	11	13	N/A	19	4	0	8	13	2	10	14	N/A
19	3	0	2	11	15	18	6	18	19	4	0	9	11	7	12	16	N/A
19	3	0	3	10	12	16	2	16	19	4	0	9	17	6	11	14	N/A
20	3	0	5	13	14	16	13	17	20	4	0	6	15	9	10	17	N/A
20	3	0	6	8	13	14	14	16	20	4	0	12	13	8	12	17	N/A
21	3	0	2	4	6	20	6	8	21	3	0	8	12	16	18	4	6
21	3	0	4	9	14	19	11	17	21	3	0	10	12	13	17	10	18

Continued on next page

Table 7.1 – continued from previous page

n	$type$	0	a	b	c	d	e	h	n	$type$	0	a	b	c	d	e	h
22	2	0	17	8	15	10	18	N/A	22	3	0	7	11	16	17	1	13
22	2	0	21	4	11	12	18	N/A	22	3	0	10	14	16	21	5	12
22	3	0	4	6	12	21	1	19	22	4	0	16	17	12	17	20	N/A
23	2	0	20	4	5	4	10	N/A	23	3	0	5	7	12	20	4	19
23	3	0	9	13	17	21	10	18	24	3	0	9	15	16	23	6	17
25	4	0	9	21	2	4	24	N/A									

REFERENCES

- [1] E. Dubrova, “A list of maximum-period NLFSRs,” 2012.
- [2] “The general structure of an n-bit lfsr,” https://awrcorp.com/download/faq/english/docs/VSS_System_Blocks/LFSR_SRC.htm, note = Accessed on 18.05.2017.
- [3] E. Dubrova, “A scalable method for constructing galois NLFSRs with period 2^{n-1} using cross-join pairs,” *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 703–709, 2013.
- [4] C. J. A. Jansen, “Investigations on nonlinear streamcipher systems: construction and evaluation methods,” 1989.
- [5] E. Dubrova, “Generation of full cycles by a composition of nlfsrs,” *Designs, codes and cryptography*, vol. 73, no. 2, pp. 469–486, 2014.
- [6] “Application of linear feedback shift register,” https://en.wikipedia.org/wiki/Linear-feedback_shift_register, note = Accessed on 28.04.2018.

- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] S. Golomb, “Shift register sequences. laguna hills, CA Aegean,” 1982.
- [12] J. Massey, “Shift-register synthesis and BCH decoding,” *IEEE transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.
- [13] 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.
- [14] 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.

- [15] H. Fredricksen, “A survey of full length nonlinear shift register cycle algorithms,” *SIAM review*, vol. 24, no. 2, pp. 195–221, 1982.
- [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, “A transformation from the fibonacci to the galois NLFSRs,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5263–5271, 2009.
- [18] F. Masoodi, S. Alam, and M. Bokhari, “An analysis of linear feedback shift registers in stream ciphers,” *International Journal of Computer Applications*, vol. 46, no. 17, pp. 46–49, 2012.
- [19] “Key derivation function,” https://en.wikipedia.org/wiki/Key_derivation_function, note = Accessed on 11.12.2017.
- [20] “Distinguishing Attack,” https://en.wikipedia.org/wiki/Distinguishing_attack, note = Accessed on 11.12.2017.
- [21] I. Ahmad, S. Alam, and M. Bokhari, “Comparative study of techniques and attacks on NLFSR based stream ciphers—a review.”
- [22] M. A. Orumiehchiha, J. Pieprzyk, R. Steinfeld, and H. Bartlett, “Security analysis of linearly filtered NLFSRs,” *Journal of Mathematical Cryptology*, vol. 7, no. 4, pp. 313–332, 2013.

- [23] C. Berbain, H. Gilbert, and A. Maximov, “Cryptanalysis of grain,” in *FSE*, vol. 6. Springer, 2006, pp. 15–29.
- [24] M. Afzal and A. Masood, “Algebraic cryptanalysis of a nlfsr based stream cipher,” in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*. IEEE, 2008, pp. 1–6.
- [25] S. Deb, B. Biswas, and N. Kar, “Study of NLFSR and reasonable security improvement on trivium cipher,” in *Information Systems Design and Intelligent Applications*. Springer, 2015, pp. 731–739.
- [26] K. Pommerening, “Cryptanalysis of nonlinear feedback shift registers,” *Cryptologia*, vol. 40, no. 4, pp. 303–315, 2016.
- [27] “Cryptanalysis of stream ciphers,” <http://www.ecrypt.eu.org/stvl/sasc/meier-sasc.pdf>, note = Accessed on 15.05.2017.
- [28] “Introduction to design and analysis of stream ciphers,” https://www.cosic.esat.kuleuven.be/summer_school_albena/slides/Introduction_Stream.pdf, note = Accessed on 15.05.2017.
- [29] E. Dubrova, M. Teslenko, and H. Tenhunen, “On analysis and synthesis of (n, k) -non-linear feedback shift registers,” in *Design, Automation and Test in Europe, 2008. DATE’08*. IEEE, 2008, pp. 1286–1291.

- [30] I. Janicka-Lipska and J. Stokłosa, “Boolean feedback functions for full-length nonlinear shift registers,” *Journal of Telecommunications and Information Technology*, pp. 28–30, 2004.
- [31] 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.
- [32] 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.
- [33] L. Zhiqiang, “The transformation from the galois NLFSR to the fibonacci configuration,” in *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 335–339.
- [34] J. Mykkeltveit, M.-K. Siu, and P. Tong, “On the cycle structure of some nonlinear shift register sequences,” *Information and Control*, vol. 43, no. 2, pp. 202–215, 1979.
- [35] J. Mykkeltveit and J. Szmidt, “On cross joining de bruijn sequences,” in *Proc. Topics Finite Field Contem. Math.*, vol. 632, 2015, pp. 335–346.
- [36] K. Mandal and G. Gong, “Cryptographic d-morphic analysis and fast implementations of composited de bruijn sequences,” Technical Report 27, University of Waterloo, CACR, Tech. Rep., 2012.

- [37] C. G. Günther, “Alternating step generators controlled by de bruijn sequences,” in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1987, pp. 5–14.
- [38] C. F. Sainte-Marie, “Solution to question nr. 48,” *L’Intermédiaire des Mathématiciens*, vol. 1, pp. 107–110, 1894.
- [39] N. G. De Bruijn, “A combinatorial problem,” in *Nederl. Akad. Wetensch. Proc.*, vol. 49, 1946, pp. 758–764.
- [40] 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.
- [41] K. Mandal and G. Gong, “Probabilistic generation of good span n sequences from nonlinear feedback shift registers,” *University of Waterloo*, 2012.
- [42] I. J. Good, “Normal recurring decimals,” *Journal of the London Mathematical Society*, vol. 1, no. 3, pp. 167–169, 1946.
- [43] R. Motwani and P. Raghavan, “Randomized algorithms. cambridge international series on parallel computation,” 1995.
- [44] 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.

- [45] M. Liu, “Analysis and synthesis of boolean networks,” Ph.D. dissertation, KTH Royal Institute of Technology, 2015.
- [46] P. Dąbrowski, G. abuzek, T. Rachwalik, and J. Szmidt, “Searching for nonlinear feedback shift registers with parallel computing,” *Information Processing Letters*, vol. 114, no. 5, pp. 268–272, 2014.
- [47] N. Poluyanenko, “Development of the search method for non-linear shift registers using hardware, implemented on field programmable gate arrays,” *EU-REKA: Physics and Engineering*, no. 1, pp. 53–60, 2017.
- [48] A. Poorghanad, A. Sadr, and A. Kashanipour, “Generating high quality pseudo random number using evolutionary methods,” in *Computational Intelligence and Security, 2008. CIS’08. International Conference on*, vol. 1. IEEE, 2008, pp. 331–335.
- [49] H. Hu and G. Gong, “Periods on two kinds of nonlinear feedback shift registers with time varying feedback functions,” *International Journal of Foundations of Computer Science*, vol. 22, no. 06, pp. 1317–1329, 2011.
- [50] B. Gammel, R. Göttfert, and O. Kniffler, “The Achterbahn stream cipher,” *Submission to eSTREAM*, 2005.
- [51] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge university press, 1994.

Vitae

Personal details:

- Name: Ibraheem Mohammed Ahmed Al-Hejri
- Nationality: Yemeni
- Date of Birth: July 1, 1987
- Personal Email: *alhejri87@gmail.com*
- Permanent Address: Thamar, Yemen

Education, Research, and Experience:

- BSc in Computer Science & Information System from Thamar University, Yemen (September 2005 - September 2009).
- Research Interests: Info. Assurance & Security, Data Encryption & Security, Network Security, Web Applications Security.
- Academic Experience: Lecturer, Thamar University, Yemen (October 2009 - August 2014).