Five Decade Evolution of Feedback Shift Register : Algorithms, Architectures and Applications

K.K. Soundra Pandian

Department of Electrical Engineering,
Indian Institute of Technology Patna,
Patna, Bihar, India
E-mail: kpandian@iitp.ac.in

Kailash Chandra Ray

Department of Electrical Engineering,
Indian Institute of Technology Patna,
Patna, Bihar, India
E-mail: kcr@iitp.ac.in

Abstract: The accomplishment and insinuation of feedback shift register (FSR) for the past five decades, lies in the simplest type of digital sequential network circuit with the unit delay can be the physical or the storage element, to deliver a sequence of binary bits, repeat after a period due to the circuit states of finite numbers. The state sequential network by framing the logic for state variable has found wide usage in the sequence or code generation, counting, and sequence recognition or decoding. Thus the logical design by shifting the registered bits from the physical or storage element exhibits the concept of shift register in the general sequential network. The shift register provided with the suitable feedback logic, capable of generating long string of binary digit possessing quasi-random properties. The high-speed communication cryptography, stream cipher, test pattern generator, image encryption, cyclic redundancy check (CRC) operation and Bose-Chaudhuri-Hocquenghem (BCH) encoder require the generation of random numbers in which the FSR have been utilized. This paper addresses a brief overview of the key expansion in the feedback shift register and feedback with carry shift register of either linear or non-linear with the sequential and/or parallel processing architectures along with their concealed and imminent applications.

Keywords: linear feedback shift register; non-linear feedback shift register; feedback with carry shift register algorithms; cyclic redundancy check; pseudo random generator; stream cipher; encryption; decryption; cryptography

Biographical notes:
K.K.Soundra Pandian is presently pursuing his PhD degree in the Department of Electrical Engineering, Indian Institute of Technology Patna, India since December 2012. He is working as Research Engineer Grade-I with the Department of Electrical Engineering, Indian Institute of Information Technology Design and Manufacturing Jabalpur (IIITDMJ), India since January 2008.
His major research area includes VLSI Cryptography, Hardware Security, FPGA Based System Design, Pseudorandom Generator, Symmetric Cipher.

Kailash Chandra Ray is presently serving as an Assistant Professor with the Department of Electrical Engineering, Indian Institute of Technology Patna, India since June 2010. Prior to this, he served as a lecturer in the Indian Institute of Information Technology Allahabad, India from August 2008 to June 2010. He worked as a Technical Member Staff (Design Engineer) for CMOS Chips Bangalore, India during March 2001 -February 2003. He received his Ph.D. degree in Electronics and Electrical Communication Engineering from the Indian Institute of Technology Kharagpur, India in 2009. He received his M. Tech degree in Electrical Engineering from the Regional Institute of Technology (Presently NIT) Jameshedpur, India in 2000 and Bachelor of Engineering Degree in Electrical Engineering from the Orissa Engineering College Bhubaneswar, India in 1997.
His research interests include VLSI architectural design, VLSI Signal Processing, Digital VLSI Design, Hardware Design Methodologies, FPGA Based System Design, CORDIC, and Embedded System.

1 Introduction

In the linear regime, the concept of the linear-product type feedback shift register (FSR) staunched from the simple hypothetical descriptions provided by a polynomial representation of the feedback function which is a generalisation of the linear type and could be applied to the general non-linear FSR to give a flexible theoretical description. *Zierler* [1] in 1955 postulates quite a few binary sequence generator from the polynomial representation of the feedback function. In same year *golomb* [2] postulates how to generate the randomness properties with the sequences generated. The main objective in deriving this polynomial technique is to set up a theoretical description for the formation of polynomial products and the factorisation of composite types. The binary sequence situation is generalized to that of a multi-valued p-binary logic, where p is any prime integer as suggested by *huffman* [3] in 1956. These techniques would then be applicable to the description of non-linear product FSR. The research on linear feedback shift registers (LFSR) started in the early 60's by *elspas* [4] in 1959 and continued actively for many years and the key concept of FSR is based on the linear sequential network as described [4] in 1959. Furthermore, the error detection using the cyclic codes is postulated [5] in 1961, for the network logical structure and state logic relations for sequential networks behaviour for all possible cycle lengths postulated [6] in 1964, not merely for maximal cycle lengths. Thus, the class of networks treated here is not limited to shift registers with feedback, but includes arbitrary interconnections of delay elements. The linear logic elements with the serial-to-parallel transformation of LFSR [7] and mapping between the equivalent states chosen to be a linear transformation postulated [8] in same year.

The collection of contralinear feedback shift register with utmost one invertor [9] in the feedback path is similar to that of linear sequence network and the logical design procedure for feedback shift registers, which permits the gating of a common clock signal [10] and the binary sequence are constructed using *Berlekamp-Massey* algorithm with the minimal LFSR generating a given binary sequence and to synthesis the LFSR with a maximum period with the primitive generator polynomial hypothesized [11] in 1969. The polynomial representation of non-linear feedback shift register with the invertors [12] and LFSR associated sequences have become increasingly more important in many areas of electronics

and communication during the past few decades.

The success of LFSR, stemmed from the simple theoretical descriptions provided by a polynomial representation of the feedback function. Based on finite fields LFSR-based computational problems to their counterparts have increased attention in 70's. Feedback logic of a non-linear type feedback shift register to be represented as a non-linear polynomial, using K-maps [13] and the polynomial form of the non-linear product feedback shift register in the sequential behaviour hypothesized [14] in 1970. Thus the comprehensive clarification of the cyclic redundancy check and double adjacencies with circular shift in feedback shift registers [15, 16, 17] in early 70's. During the mid and end of 70's, *fredricksen* suggested the non-linear *k-ary debruijn* cycle of sequences which is the stepping stone for the non-linear sequences [18, 19, 20] and *mykkeltveit* [21, 22] have postulate the non-linear recurrences and arithmetic codes for the feedback shift registers. Numerous research have been adopted with the debruijn sequences and developed the memory less algorithm [23] in 1981 for the feedback shift register binary sequences. FSR numerous applications, including cipher system for the secure commutations suggested [24] in 1982, testing [25] in 1985, stream cipher cryptography [26] in 1991, error detection and correction [27] in 1999 and data compression [28] in 2004. Now-a-days LFSRs are well understood and grown-up in the global market and most fundamental problems of characteristic polynomial, linear complexity related to LFSRs are solved.

1.1 Linear Feedback Shift Registers

The better choice for pseudo-random number sequence generators and its applications is the feedback shift register technique. As in a normal shift register in the fibonacci implementation as shown in Fig.1, all the logic bits except the last is shifted each cycle. Due to the non-linear function of the preceding bits, the last bit is updated accordingly. As generalization of the fibonacci implementation, it can be considered as galois implementation as shown in Fig.2. According to the next-state function, each bit in a galois implementation is updated, which is a non-linear function of the previous bit which is up to k^{th} other bits, this is obvious that galois architecture is faster than fibonacci architecture, since the propagation time for smaller function of the individual bits in galois configuration is reduced as compared with the large feedback function in fibonacci configuration.

Figure 1 : Fibonacci LFSR

As a significant importance, LFSRs are commonly used in very large scale integrated circuit testing and other non-secure applications. A shortest stage of binary machine for generating a given periodic sequence, an algorithm [29] is constructed using one or more linear product feedback shift registers by combining the states of LFSRs or by clocking the LFSRs in a modified manner (A5 cryptographic key generator) [30]. Thus, the combining based scheme takes the outputs of several LFSRs to produce the NLFSR output. The main drawback of LFSRs in the context of cryptography is their linearity through cryptanalysis and the structure of a n-bit LFSR can be inferred by observing consecutive bits of its sequence [31]. Therefore, LFSRs cannot directly be used in cryptography applications.

Figure 2 : Galois LFSR

The NLFSRs with non-linear feedback functions, either fibonacci or galois implementations are used to overcome the drawback in linear feedback shift register which are commonly used and be more secure than LFSR in the cryptanalysis attacks [32]. Statistical properties of the binary sequences by LFSR have been analysed to satisfy postulates of *golomb's* [33] $1st$ and $2nd$, by which the generated random number sequences are of maximum length. It is extremely difficult task to construct the feedback function for large NLFSRs with guaranteed maximal length sequences, using smallest register with non-linear feedback function [34]. The basic comparison of fibonacci and galois LFSR is summarized in Table.1.

Table 1 : Comparison of Fibonacci and Galois LFSR

1.2 Non-Linear Feedback Shift Registers

An excellent overview of the full length non-linear shift register cycle algorithm [35] in which, the binary bit sequence of NLFSR are a generalization of linear product FSRs. Next, the non-linear function of the previous states are generated by the current state by constructing a minimal NLFSR algorithm generating a sequence of binary state presented [36] in 1989. Systematic procedure for constructing modified NLFSRs with a bit sequence of long period [37] with the existing algorithms either considering certain special cases postulated by *dubrova* [38]. Thus, it is applicable to small NLFSRs only, is suggested by *frederickson* [39, 18, 19, 20] and other applicable existing algorithms for generating the best possible sequences to NLFSRs postulated [40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 30, 51] in detail. The non-linear feedback shift registers used for the security techniques (cryptographic key generation) utilizes the non linearity feedback function for the next state to generate the maximal length sequences output.

For NLFSRs, the fibonacci to galois transformation is not unique [52, 53, 54], to transform from fibonacci to galois NLFSRs configuration can potentially reduce the depth of the circuits by implementing logical feedback functions, thus increasing the throughput and decreasing the propagation time [53]. The mathematical properties of LFSRs can be leveraged to derive analytical guarantees of the NLFSR performance. The clock controlled NLFSRs use irregular clocking for one or more of the LFSRs, based on some sub-generator or functions of the LFSR states. In the security techniques method, the $A5$ cryptographic key generator consists of three LFSRs, in which a specific clocking bit from each LFSR is used to calculate a majority output. The clocking of each LFSR is done only if its clocking bit contests with the majority output and the output of the $A5$ cryptographic key generator is the XOR of the last bit of each LFSR. Thus the general drawbacks are eliminated and reduced with clock-controlled periods schemes and hence improve the linearity of the NLFSR output.

This paper conferred the principles of linear type feedback shift register operation, covering the elementary ideas from composition of the characteristic polynomial of the transition matrix determines the behaviour of the FSR and sequential operations can be explained for the periods, cycle sets and sequence structures, when the characteristic polynomial is either irreducible or composite followed by design of basic linear and non-linear FSR.

The remainder of this paper is organised as follows, the key developments in FSR algorithms and architectures are conferred in Section II, which covers the algorithms and architectures pertaining to analysing feedback functions of maximum period NLFSRs and by choosing a sequence of states, a heuristic approach minimizes the gate complexity of the next state

functions. This confer the key developments of fibonacci and galois feedback with carry shift registers algorithm and architectures with the space representation and basics of the linear FSR techniques either serial or parallel type architectures. The necessity of feedback shift register applications and brief classification are discussed in Section III. The applications of FSR are conferred briefly in Section IV. The conclusion along with future research directions are discussed in Section V.

2 Algorithms and Architecture

In this section, the basic properties and algorithm of LFSR sequence are conferred, from composition of the characteristic polynomial, however the sequential operation can be explained for the periods, cycle sets and sequence structures, where the characteristic polynomial is either irreducible or composite followed by design of the basic linear and non-linear FSR.

2.1 The FSR Algorithm

The feedback shift register (FSR) represented in normalized form of order *n* as shown in Fig.3 with the storage element in binary format called stages or bits. Each associated state variable a_i represents the current value of the stage i where $i \in \{0, 1, \cdots, n-1\}$ and a feedback function $f_i : \{0,1\} \quad n \rightarrow \{0,1\}$ determines the value of i is being updated. A state of an FSR is a vector of values of its state variables $(a_0, a_1, \cdots a_{n-1})$. The current state of FSR is determined from the next state of FSR at every clock cycle, simultaneously updating the value of f_i of each stage i. The input fed to FSR is the value of its $(n-1)$ stage as referred to equation (1). The longest cyclic output sequence length determines the period of FSR and the output value of its stage 0 is produced. FSRs is of type linear, then all the feedback functions, i.e., of type

$$f(a_0, a_1, \cdots, a_{n-1}) = c_0 \oplus c_1 a_0 \oplus c_2 a_1 \oplus \cdots c_n a_{n-1} \tag{1}$$

where $c_i \in \{0,1\}$ for $i \in \{0, 1, \cdots, n\}$, then it is called a linear feedback shift register.

Figure 3 : General Structure of FSR

2.1.1 Properties of FSR sequences

The operation of a FSR is largely determined by its characteristic polynomial over the binary sequence which is closely related to the output sequence properties [51]. Some of the properties for the maximum length of binary sequence are summarized below,

$Property - 1) : Characteristic\ Polynomial$: The connection polynomial corresponds to r-stage LFSR stages as in equation (2), with the r taps $(p_1, p_2, \cdots, p_r)$ on the cells of an r-stages.

$$p(X) = p_r X^r + p_{r-1} X^{r-1} + \cdots + p_1 X - 1 \qquad where \quad r \in \{0, 1, \cdots, n\} \tag{2}$$

The output sequence $p(X)$ of a LFSR is periodic that depends on both the initial condition and the characteristic polynomial. The period of the LFSR sequence expressed in terms of the characteristic polynomial.

$Property-2): Power\ Series:$ The binary sequence $a=(a_0,a_1,a_2,\cdots)$ which are infinite, is identified by its generating function as referred to equation (3),

$$A(X)=\sum_{i=0}^{\infty}X^i \tag{3}$$

in which the formal power series with integers modulo 2 with the coefficients are deterministic, with the ring $Z/2\ [[X]]$ consist of an element. The sequence a is eventually periodic as shown in equation (4), the quotient of two polynomials are generated from the function,

$$A(X)=(r(X)/p(X))\in Z/2\ [[X]] \tag{4}$$

in which the numerator $r(X)$ is the initial seed polynomial and the periodic part of the sequence a generated by the linear feedback shift register where denominator $p(X)$ is the connection polynomial of it.

$Property-3): Galois\ Field:$ The linear feedback shift register with connection polynomial $p(X)$ of length r, let us consider $a=(a_0,a_1,a_2,\cdots)$ is a sequence of bits which is periodic and irreducible if $\gamma\in GF(2^r)$ is a root of $p(X)$ for all $i=0,1,2,\cdots,$ where $a_i=Tr(A\gamma^i)$ for $A\in GF(2^r)$ which resembles the initializing of shift register with the choice of initial seed loading. Hence, galois field denotes the function to trace $Tr:GF(2^r)\rightarrow GF(2)$.

$Property-4): Periodic:$ The smallest size of LFSR generates a sequence a which is of periodic in regular which is termed as linear complexity of a and it is the significant measure of the security application in cryptographic. The *berlekamp-massey* algorithm provides the efficient way to design the minimum size shift registers. This algorithm in two senses can be optimal (i) it decide the periodic sequence a coincides with the smallest LFSR and (ii) minimal information i.e., i^{th} bits sequence of first two spans are needed.

$Property-5): Maximum\ Length:$ The maximal length sequence (also known as m-sequence) is a LFSR sequence have maximum possible period $T=2^n-1$ for n stages, in which LFSRs generates the m-sequences exactly those sequences corresponds to the taps of connection polynomials which is primitive. Thus the sequences are well balanced in the any period of an m-sequence which is single and have the *deBruijn* property [55], more over every non zero binary string of length r befalls exactly once.

The cyclic *reed-muller* code of first order, have the 2^n-1 cyclic permutations of a period of an m-sequence which is single, form the non-zero code words. The fundamental significance of these codes is in the area of coding theory and are patterns of the general *finite geometry* codes.

2.2 *The Cyclic Redundancy Check (CRC) and Bose-Chaudhuri-Hocquenghem (BCH) Algorithm*

For CRC and BCH generator polynomials $d_K = d_0 = 1$, in GF(2) as referred to equation (5). Let, for $n = (0, 1, \cdots, N-1)$, represent an input sequence of length N.

$$g(x) = \sum_{n=0}^{N-1} g(n)x^n, \quad where\ g(n) \in GF(2) \tag{5}$$

To attain the remainder $(g(x)x^K)_{d(x)}$ by the division of the polynomial $g(x)x^K$ by $d(x)$ is encompassed by the CRC computation and BCH encoding [56]. The open or short circuit connection exists for implementing two-input XOR gate and the multiplier elements in which two elements are added, its implies that no connection exists and the direct connection from input to output can be replaced from the corresponding XOR gate. The N-bit message is fed to the most significant bit of LFSR, during the initial N clock cycles. To form the BCH encoded code word, the feedback is reset to zero, simultaneously the bits of message are fed to the output after every N clock cycles and the K registers contains the coefficients of reminder $(g(x)x^K)_{d(x)}$.
In BCH encoding, to form the code word bits in systematic, the remaining bits are shifted out bit by bit in sequence. The system throughput is modifies to increase the process for the number of bits in parallel and for the serial computation of the LFSR is limited. In the linear feedback shift register of serial to parallel transformation [7] are described and first applied to computation of CRC [15]. Several other methodologies have been recently presented to parallel LFSR computations. For stream ciphers, it is conferred that, the basic principle underlying the FSR-based algorithm for different pseudo-random sequences. The galois field $GF(2^m)$ specifies the binary BCH code generator polynomial, where m $(m \geq 3)$ is a positive integer. Let α be a primitive element in $GF(2^m)$ galois field. For all the values of i, $m_i(x)$ be the minimal polynomial of α^i with coefficients in $GF(2)$. The BCH code generator polynomial is defined as least common multiple (LCM) [33] of $m_1(x), m_2(x), \cdots, m_{d-1}(x)$ i.e.,

$$d(x) = LCM(m_1(x), m_2(x), \cdots, m_{d-1}(x)) \tag{6}$$

where $d \leq 2^m - 1$, $d(x)$ as shown in equation (6) has a distinct root and their conjugates as all its roots [57].

Theorem 1: In $GF(q)$ of degree k, the function f is an irreducible polynomial then f has a root α in $GF(q^k)$. For the k distinct elements $\alpha, \alpha^q, \alpha^{q2}, \cdots, \alpha^{qk-1}$ of $GF(q^k)$, the roots of f are simple. The generator polynomial $d(x)$ as shown in equation (7) of CRC code is generated by either a primitive polynomial or a polynomial

$$d(x) = (x+1)p(x) \tag{7}$$

where $p(x)$ can be product of one or more primitive polynomials [58]. Any primitive polynomial is irreducible and its extension field has distinct roots. The generator polynomial with distinct roots of $1 + x$ is 1.

2.3 *FSR Architecture*

FSRs are attractive due to their simple implementation in hardware, fast performance and it possess several appealing mathematical properties. For example, FSRs of length n whose characteristic polynomials yield maximum length sequences (m-sequences) cycle satisfies the relationship m = 2^n-1, through all non-zero states before repeating a state.

2.3.1 *Linear FSR Architecture*

Each bit of the LFSR state exhibits a uniform probability of $0's$ and $1's$, in a random sequence [59] and can be either implemented in the galois or fibonacci configuration where it exists a unique transformation between the galois and fibonacci configurations either serial or parallel architecture. The fibonacci configuration can be obtained from the galois one (and vice verse) by reversing the order of LFSRs feedback taps and by adjusting the initial state.

2.3.1.1 *Serial Linear FSR Architecture*

Let consider that, the input to the LFSR is $u(n)$ as shown in equation (8), and the required output $z(n)$, i.e., the remainder is $y(n)$ as referred to equation (10) as shown in Fig.4. Thus the following equation describes the LFSR architecture,

$$z(n) = y(n) + u(n) \tag{8}$$

$$y(n) = d_{K-1} * z(n-1) + d_{K-2} * z(n-2) + \cdots + d_0(z(n-K) \tag{9}$$

Substituting equation (8) into equation (9), we get

$$y(n) = d_{K-1} * y(n-1) + d_{K-2} * y(n-2) + \cdots + d_0 * y(n-K) + f(n) \tag{10}$$

where

$$f(n) = d_{K-1} * u(n-1) + d_{K-2} * u(n-2) + \cdots + d_0 * u(n-K) \tag{11}$$

The function $f(n)$ as referred to equation (11) resembles an filter response with $(d_0, d_1, \cdots, d_{K-1})$ as coefficients in which $'+'$ denotes the XOR operation. The generating polynomial order represents the basic LFSR as shown in Fig.4 represents the LFSR length i.e., the coefficients of the characteristic polynomial is represented as the delay elements. The characteristic of polynomial $d(x)$ as shown in equation (12) of the LFSR is:

$$d(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_K x^K \tag{12}$$

where $(d_0, d_1, d_2, \cdots, d_K) \in GF(2)$. In order to find which tap positions will yield maximal length sequences, must enter the realm of finite field $F_2[x]$, where all the binary field operations are done using modulo 2. The characteristic generating polynomial or feedback

Figure 4 : General LFSR Architecture

polynomial as shown in equation (13) of an LFSR, is the polynomial so that the tap positions are correlated,

$$d(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x^1 + a_0 \tag{13}$$

The a_n coefficients can either be $1's$ or $0's$ and in all LFSR applications, a_0 must be 1, which refers to the fact that there is feedback fed into the new state of LFSR. It is known that if $d(x)$ divides $x^N - 1$, where $N = 2^n - 1, N$ is prime, then is said to be primitive, and maximal length sequence correlates to tap positions, satisfy the *golomb* postulates of $1st$ and $2nd$. For example, for $n = 3$, let consider $q(y) = y^7 + 1$. The polynomial factor as shown in equation (14) considered as follows,

$$q(y) = y^7 + 1 = (y + 1)(y^3 + y^2 + 1)(y^3 + y + 1) \tag{14}$$

To create a maximal length sequence of 7 registers long, the last two factors are $1s$. The degree of $(y + 1) = 1$ and $d(x)$ will not provide the maximal length tap positions. Therefore, for the LFSR of 3-bits indicates the maximum tap positions are $(1, 3)$ and $(2, 3)$. Thus for larger n-bit registers process can be very monotonous by which the tap positions for LFSR of larger length are well known, and thus the maximal tap and non-deterministic positions of taps can be determined by the scientific progression. Let $d(x)$ represents the characteristic polynomials, that produce the maximal length sequences are essentially irreducible polynomials over the galois field element $GF(2^n)$. If all the elements from a galois field using a finite binary field $F_2[x]$ (i.e., finite binary field $F_2[x] \,/\, d(x)$) and characteristic polynomials $d(x)$ are associated to the account of all states using $d(x)$ as the characteristic polynomial of LFSR, the order of the states sequence generated may not be of the same sequence, will be alike.

A set of tap positions corresponds to the mirror image, will produce a mirror image sequence thus the non-maximal length tap positions correlate to reducible polynomials, it is trivial to determine the next irreducible polynomial. The reversed feedback set is described by $[n, n - C, n - B, n - A]$ for the original feedback set of $[n, A, B, C]$, where n is the number of LFSR stages. Thus, if the given one irreducible polynomial is $GF(2^n)$, it is trivial to determine the next irreducible polynomial. The information that each register state of LFSR is an element of galois field $GF(2^n)$ over the characteristic polynomial $d(x)$. The relationship of LFSR and galois fields can be exploited into Advanced Encryption Standard (AES) of cryptographic application.

The statistical modelling [36] portray the behaviour of the maximum complexity, $\eta(c_i)$ denote the probable number of increasing length and the maximum order complexity (c_i) of a random sequence as shown in equation (15) of length l,

$$\eta(c_i) = 2 \log_b l \tag{15}$$

In order to construct the feedback shift register of shortest length sequence, initially requires the maximum order complexity and its feedback function are determined. The *deBruijn* sequences have a maximum of order n complexity [35] and non-random which generates a given sequence of length l. The initial order operation is proportional to l and is expected to produce $2 \log l$ complexity value and the next operation with standard technique can be performed [36], has order $c2^c$ for the binary case. Hence, the expected order of the FSR synthesis procedure is $2l^2 \log l$ as opposed to $l \log l$ for *deBruijn* sequences. The standalone

FSR are insecure and structured with characteristic polynomials, can be easily recovered from the output sequence by (i) *Berlekamp-Massey* [60] Algorithm and (ii) *Extended Euclidean* Algorithm [61]. The feasibility of general FSR resynthesis limits the expected order to moderate the length of sequences.

2.3.1.2 ***Parallel Linear FSR Architecture***

A parallel linear FSR architecture based on computation of state space transformation [62, 56]. The LFSR as shown in Fig.4 as referred to equation (16) can be described by

$$y(n+1) = \mathrm{D}y(n) + \mathrm{e}u(n); \qquad n \geq 0 \tag{16}$$

with the initial state $y(0) = y_0$. The K-dimensional state vector $\mathrm{y}(n)$ as shown in equation (17),

$$y(n) = [y_0(n) \;\; y_1(n) \;\; \cdots \;\; y_{K-1}(n)]^{\mathrm{T}} \tag{17}$$

and D is the K x K matrix given by,

$$\big[D\big] = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & d_0 \\ 1 & 0 & 0 & \cdots & 0 & d_1 \\ 0 & 1 & 0 & \cdots & 0 & d_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & d_{K-1} \end{bmatrix} \tag{18}$$

The K x 1 matrix *e* as shown in equation (19),

$$e = [d_0 \quad d_1 \quad \cdots \quad d_{K-1}]^T \tag{19}$$

The remainder of the polynomial division computes the state vectors which determines the system output. The output vector $z(n)$ is added to the output equation $z(n) = Dy(n)$ to the state equation, in which A equal to the K x K matrix which is identity as shown in equation (18). The coefficients of the generator polynomial $d(x)$ appear in the right-hand

Figure 5 : Parallel LFSR Architecture

column of the matrix D. This is the companion matrix of characteristic polynomial $d(x)$, the initial state y_0 depends on the given application.
In broad-spectrum, parallel architectures of LFSR as shown in Fig.5 proposed to an increase in the critical path. The operation speed of the circuit get decreased as the critical path is longer, thus by parallel processing the throughput rate will be reduced. To decrease the critical path, the feedback loops of parallel architectures and pipe lining technique cannot be applied due to the delay elements and other issue is the hardware cost. The novel parallel CRC architecture [62] based on the state space representation used to shift out the feedback loop complexity. To increase the speed-up, based on state space transformation can be achieved by pipe-lining the feed forward paths. In the parallel system architecture, block delay is referred by the each delay element where the original sample bit period is achieved from the clock period of parallel system which is of three times the sample bits.

The L-parallel LFSR architecture [62] system are processed in parallel, with the L elements of the input sequence $u(n)$. The input to the parallel system is a vector $u_L(wL)$ as shown in equation (20), where N (integral multiple of L) is the length of the input sequence.

$$\begin{aligned} u_L(wL) = [u(wL+L-1)u(wL+L-2)\cdots \\ u(wL+1)u(wL)]^T; \\ where \quad w = 0,1,\cdots\left(\frac{N}{L}\right)-1 \end{aligned} \tag{20}$$

The state space equation can be written as

$$\begin{aligned} x(wL+L) = D^L y(wL) + E_L u_L(wL); \\ z(wL) = C_L y(wL); \end{aligned} \tag{21}$$

where the index wL is incremented by L for each block of L input bits. The matrix E_L as shown in equation (22) is a $K \times L$ matrix given by

$$E_L = [e \;\; De \;\; D^2e \;\; \cdots \;\; D^{L-1}e]; \tag{22}$$

and C_L is $N \times N$ identity matrix. The output vector $z(wL)$ as referred to equation (23) is equal to the state vector which has the remainder at

$$m = \left(\frac{N}{L}\right) \tag{23}$$

The L-parallel system which processes L-bits at a time as shown in Fig.5, which possess the delay issue in the feedback loop, increases due to the complexity of D^L. In high-speed parallel architecture [56] for LFSR, the block delay is referred by each delay element, the throughput of the system is limited by delay in the feedback loop.

2.3.2 Non-linear FSR Architecture

The Non-Linear Feedback Shift Register (NLFSR) [52] consists of bits of storage elements i where $i \in \{0,1,\cdots n-1\}$, associated with state variable is represented as a_i which represents the current value and it determines the updated values of bit i for the feedback function f_i. For any value of i, f_i depends on $a_{(i+1)} mod\ n$. The NLFSR state is a values of its state variables $a = (a_0, a_1, \cdots, a_{n-1})$. The next state of NLFSR is determined from the current state at every clock cycle, by simultaneously updating the function f_i for the value of each bit i. Feedback functions of NLFSRs are represented in the algebraic normal form (ANF) which is a polynomial in the form of galois field $GF(2)$ type

$$f(x) = \sum_{i=0}^{2^n-1} b_i \cdot a_0^{i0} \cdot a_1^{i1} \cdot \cdots \cdot a_{n-1}^{i_{n-1}} \tag{24}$$

where $b_i \in \{0,1\}$ and $(i_0, i_1, \cdots i_{n-1})$ is the binary expansion of i. The boolean function $f(x)$ dependence set is shown in equation (24),

$$debf(f) = \{i \mid f\ |_{ai=0} \neq f\,|_{ai=1}(x)\} \tag{25}$$

Figure 6 : An n-bit Fibonacci NLFSR Architecture

where

$$f\,|_{ai=j} = f(a_0, \cdots a_{i-1}, j, a_{i+1}, \cdots, a_{n-1}) \quad for j \in \{0, 1\} \tag{26}$$

The equivalence conditions of two NLFSRs [53] are equivalent if the binary output sequences sets are equal. NLFSR can be implemented in two configurations like LFSRs, either fibonacci configuration with external feedback as shown in Fig.6, or galois configuration with internal feedback as shown in Fig.7.

Figure 7 : An n-bit Galois NLFSR Architecture

The feedback functions of a fibonacci NLFSR, except f_{n-1} are of type $f_i = a_{i+1}$ in the fibonacci configuration, can be applied to left most bit from any bit. The feedback functions in the galois configuration [29], can be applied from any bit i to any bit j such that $j \geq i$ where $i \in \{0, 1, \cdots, n-1\}$. The feedback function of a fibonacci NLFSR is limited by the depth of the circuit due to the operating speed. To increase the operating speed, the transformation to an equivalent galois configuration from fibonacci configuration [38] of an individual bits, potentially reduces the depth of the circuits implementing with feedback functions.

Table 2 : Comparison of Fibonacci and Galois NLFSR

An algorithm for constructing a fastest galois feedback shift register of non-linear product type [52], apart from the fibonacci and the galois configurations, there are also other types of NLFSRs [63] in which there are many n-bit galois NLFSRs which are equivalent to a given n-bit fibonacci NLFSR. Further, method to generate full length sequence by NLFSR is proposed by *Dubrova* [64]. On the other hand, equivalent n-bit fibonacci NLFSR for every n-bit galois NLFSR are not very equal, by which every n-bit fibonacci NLFSRs output sequences are described by a recurrence of order n for non-linear, such a recurrence does not always exist for n-bit galois NLFSRs [65]. The brief comparative analysis of fibonacci and galois configuration type NLFSR is presented in Table.2, which shows that galois NLFSR is the obvious choice for the propagation time, which is reduced due to the parallel bit computation.

2.3.3 Feedback With Carry Shift Register Architecture

As an alternative to LFSR to design the stream ciphers, the feedback with carry shift registers (FCSRs) been proposed [66] to provide with the good statistical properties and built-in non-linearity sequences with the known period. FCSR have been classified depends upon the inputs either multiple or one common inputs along with the feedback functions. The configuration with a single feedback function which depends on multiple inputs termed as fibonacci FCSR. The configuration with the multiple feedback functions with one common input termed as galois FCSR.

2.3.3.1 Fibonacci FCSR Architecture

The FCSR as shown in Fig.8, is similar to that of a LFSR. The shift register and a feedback function are common to LFSR and FCSR, the difference is that a FCSR has a carry register are added together with the bits, instead of XOR_{ing} in case of LFSR, all the tap sequence with the bits. The carry register content from the output result of mod 2, become the new

Figure 8 : Fibonacci FCSR Architecture

bit result divided by 2. Table.3 highlights the basic comparison of LFSR and FCSR for the equivalent parameters, i.e. a feedback polynomial $q(X)$ of degree n and hamming weight k for the LFSR, and a connection integer q of bit length $n+1$ and weight k for the FCSR. The basic shift register with the secondary memory m of non-negative integer in the FCSR architecture [67] have the tapped cells contents ($0's$ or $1's$) are added as integers to the memory current contents to form an integer sum σ. The new value of the memory retained for the higher order bits $[\sigma/2]$, while the parity bit $\sigma \ (mod\ 2)$ is feedback into the first cell of the shift register. The current states $(b'_0, b'_1, \cdots b'_{r-1}; m')$ as referred to equation (27) are related to the previous states $(b_0, b_1, \cdots, b_{r-1}; m)$ given by

$$b'_i = b_{i+1}\ for\ 0 \leq i \leq r-2$$

$$2m' + b'r = m + \sum_{i=1}^{r} q_i q_{r-i} \tag{27}$$

Initial non-negative memory [37] for any value m, the memory will be decreased exponentially which lies within range of range $0 \ \leq m \ \leq \ wt(q+1)$. The number of 1's in the binary expansion of the non-negative integer x is denoted by $wt(x)$ due to this reason, the memory overflow will never occur. The memory bits at least $1 + [\log 2(wt(q+1))]$, adopt $q_r \ \neq 0$ and the connection integer defined as referred to equation (28) is fitted out with FCSR,

$$q = q_r 2^r + q_{r-1} 2^{r-1} + \cdots + q_1 2 - 1 \in \mathrm{Z} \tag{28}$$

The formal power series acquaintance to (α) as shown in equation (29) for any infinite binary sequence $\mathrm{b} = (\mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2, \cdots)$ element given by,

$$\alpha = \sum_{i=0}^{\infty} b_i 2^i \tag{29}$$

One more type of FCSR i.e. d-FCSR as shown in Fig.9, in which every carried bit is delayed by $d-1$ stage before being added to $\mathrm{Z}[\beta]$, which is in need to consist of polynomials in β with the integer coefficients with respect to the relation $\beta^d = 2$, and it is bifurcated into two parts. The part labelled $\prod$ shown in Fig.9 is an adder in $\mathrm{Z}[\beta]$. The part labelled Σ adds

Figure 9 : Fibonacci d-FCSR Architecture

the 0, 1 inputs as integers and outputs the result σ' rendering to its binary extension.

2.3.3.2 *Galois FCSR Architecture*

The FCSR in galois representation [68] form is shown in Fig.10. The bits $(q_1, q_2, \cdots, q_r)$ are multipliers, assume $q_r \neq 0$ and the memory cells or carry bits are denoted as $c_1, c_2, \cdots, c_{r-1}$. The full adder represented as $\sum$ sign, at the j^{th} adder, the following input bits are received: i) the preceding cell is represented as b_j ii) the feedback line from $b_0 q_j$ iii) the memory cell from c_j, during the next clock cycle, cells are added to form a sum σ_j $(\text{with } 1 \leq j \leq r-1)$, it is passed on to the next cell in the register, the current states b'_{j-1} as referred to equation (30) and c'_j as referred to equation (31) given by,

$$b'_{j-1} = \sigma_j \ mod\ 2 \tag{30}$$

Figure 10 : Galois FCSR Architecture

and the memories are replaced with the higher order bit,

$$c'_j = \sigma_j \text{ div } 2 \tag{31}$$

$$2c'_j + b'_{j-1} = b_0 b_j + b_j + c_j, \quad \text{for } 1 \leq \text{j} \leq \text{r} - 1 \tag{32}$$

The behavioural analyses of galois architecture circuit with reference to equation (33) for the connection integer, with reference to equation (33) the power series method used to analyse is given by,

$$q = -1 + q_1 2 + q_2 2^2 + \cdots + q_r 2^r \tag{33}$$

In the galois architecture for a d-FCSR as shown in Fig.11, before being feedback the carried bits are delayed by $d-1$ state, the register cell b_{i+d-2} input is fed by the memory or carry cell c_i, which are numbered b_0 starting from. If there are r carry cells $(c_1, \cdots, c_r)$ and r feedback multipliers $(q_1, \cdots, q_r)$, since c_r is feed into b_{r+d-1} then $r+d+1$ register cells $(b_0, \cdots, b_{r+d-2})$ are manifestly needed. The size of carry register must be at least $\log_2 t$, where t is the number of taps is well comprehensive in the architecture of galois and fibonacci [67] architecture representations of FCSR registers with the repeating period, before an initial transient and the maximum period sequence not equal to $2^n - 1$, where n is the length of the shift register sequence. The maximum period is related to the integer numbers is $q-1$ gives the taps, where q is the connection integer number as referred to equation (34).

$$q = 2q_1 + 2^2 q_2 + 2^4 q_4 + \cdots + 2^n q_n - 1 \tag{34}$$

The $\log_2(t) + \log_2(m) + 1$ steps are sufficient for the initial run of FCSR, where m represents the initial memory and t indicates the number of taps. Within the length of the FCSR of n-bits, it degenerates the continuous stream of $0's$ or $1's$. The stream cipher key is represented by the initial state of a FCSR-based key generator of which is having a set of insecure and weak keys. The novel and attractive features for using the FCSRs

Figure 11 : Galois d-FCSR Architecture

for cryptography is being pioneered [49, 51] [69]. Any prime numbers p in the $p-adic$ number system extends to the rational numbers of ordinary arithmetic, which differ from the rational number system of complex system and real. The analysis of LFSRs is based on the primitive polynomials of $mod\ 2$ addition and FCSRs is based on addition over 2-*adic* numbers as referred to equation (35). Using the *berlekamp-massey* algorithm, the theory of *2-adic* analog defines the linear complexity of 2-adic analog, which gradients the probable stream ciphers.
The main drawback of FCSRs are as insecure as LFSRs, due to the FCSR configuration structure, the automation of length r can be easily recovered from $n = 2r + 1$ bits of the output using (i) *Klapper and Goresky* algorithm [50] and (ii) *Extended Euclidean* algorithm [61] to the integers 2^n and the theory behind a LFSR, involved with the multiple carry registers can do with FCSR for the key stream sequence generation.

$$C_n = \sum_{i=0}^{n-1} c_i 2^i \tag{35}$$

The ramified extensions of the 2-*adic* numbers is based on addition due to the analysis of sequence generators [47]. The brief comparative analysis between LFSR and FCSR as presented in Table.3, which shows that FCSR is an obvious choice for attack, though it consumes bit more than hardware.

Table 3 : Comparison of LFSR and FCSR

3 Necessity of Feedback Shift Registers

The linear feedback shift register is insecure for the strong cryptographic application, as the structure of n-bit linear feedback shift register can be easily realized by perceiving $2n$ sequential bits using Berlekamp-Massey algorithm [60]. Even though the linear feedback shift register satisfies the statistical properties, but are not necessary for the stream cipher to be cryptographically strong and secure. However, due to this essential linearity characteristic, LFSR based pseudorandom generator for the cryptographic applications are prone to attacks such as algebraic [70], correlation [71], plaintext [72], timing [73]. In order to generate cryptographically robust pseudorandom sequence, more than one linear feedback shift registers with other methods are utilized to generate the non recursive pseudorandom sequences with non linear characteristics and satisfies the statistical properties.
Feedback shift register is utilized to generate the random number generators of either linear or non linear recursive for the pseudorandom test pattern generators required for numerous applications such as financial and business data, secret codes, communication systems, etc., together with the cryptographic, data security and transmission. Moreover, the brief application of linear or non-linear product type related to that of FSR is to generate the pseudorandom numbers for the image encryption and decryption, data compression, error detection and correction during transmission, test pattern and high-speed cryptography

applications etc. The next section highlights the applications of FSR to generate the non recursive pseudorandom sequences.

4 Applications of Feedback Shift Registers

High amalgamation of SoC design is challenging in number of ways using the genetic evolutionary algorithm method, randomisation method etc. Feedback shift register techniques is basically applied for the pseudo-random test pattern generators [74], are deployed to perform various application such as noise generation, error detection, mobile telephony, cryptography, data compression and error correction to fulfil the basic requirements such as fault coverage, minimized hardware complexity and automatic technique for synthesis to generate hardware. The minimization of linear dependencies can be controlled by selecting appropriate primitive characteristic polynomials and reordering the FSR cells. The rest of this section highlights few real time applications of FSR.

4.1 Random Number Generators

To generate linear recurrences, the random number generators based on modulo base 2 over binary field are typical and very suitable for simulation, because of high speed. A better sequence of numbers can be increased and improved by picking a larger LFSR and using the lower bits for the random numbers, which satisfies the statistical properties for cryptographic applications [75]. The random number generator produces a sequence of number which lacks any pattern appears to be random. Several methods for generating the randomness for various applications led to different methods. Few random number generators architecture and its application are elucidated below.

4.1.1 Mersenne Twister Generator

The pseudo-random generator based on a linear recurrence of matrix type is the MT generator [76] over a F_2 finite binary field.

Figure 12 : MT19937 Hardware Architecture

Many flaws are rectified during the fast generation of high quality pseudo-random number generators, such as statistical tests for randomness, hardware cost and relative difficulty of replicating the sequences. MT generator of various bit length word generates different sequences, where MT19937-64 with 64-bit word length and MT19937 with 32-bit word length. The uniform distribution [77] in the range of $[1, 2^k - 0]$ for the k-bit word length of MT generator have three key block with 32-bit word length in the MT19937 architecture [78] as shown in Fig.12 are the extract and generate numbers and to initialise the generators executed in parallel to reduce the increasing hardware overhead and enabling the parallelisation. Further to minimize the hardware complexities of design, modulo operations are to be replaced by the logical operations.

4.1.2 Multiply-With-Carry Generator

For High speed simulation, the pseudorandom generator exhibits certain properties are a)High dimension good structure, b) for large value of d, exhibits uniform distribution of

d-tuples, c) exhibit enormous period and d) for base b which is a power of 2, should be computable preferably.

Figure 13 : Add-With-Carry Generator

Add-with-carry (AWC) generators [79] as shown in Fig.13 satisfy certain condition to achieve the good distribution properties of *d*-tuples should be less than lag. The spectral test [80, 81, 82] for large value of *d* gets failed for the AWC generators. The MWC generator[83] and independently motivated by the cryptography [84, 47, 51] called a feedback with carry shift register as shown in Fig.10. The modified AWC generator proposed by the MWC generator satisfies the certain conditions of c) and d). The distribution properties are not optimal for the MWC sequences. The fast sequences generation of random numbers with the huge periods due to the MWC methods, raise the integer arithmetic is the main advantage of MWC generator.

4.1.3 XorShift Generator

Pseudo-random number generators using Xorshift category [85], it transforms the logical gate XOR on a number with a shifted bit. It is a bit vectors state represented by the xorshift generator. The next state is achieved by xorshift operation over a certain number at each step to the x-bit blocks in the present state, where x= 64 or 32 bit. The xorshift operation is defined as to replace the x-bit block by a xor bitwise operation of the original block by a position either to the left or right.

4.1.4 Well Equidistributed Long-period Linear (WELL) Generator

The long-period PRNGs are widely used in the WELL algorithm which is better than the mersenne twister. The MT [77] and LFSRs with huge period lengths, the states are represented over a large number of bits which requires operation on 64-bit or 32-bit from one state to next state. The WELL architecture [86] as shown in Fig.14, consists of address, generate, control, 6R/2W random access memory and temper unit.

Figure 14 : WELL Hardware Architecture

The read/write addresses for the RAM is supported by the address unit. The temper and generate operation is computed by generate and temper unit and it is fully pipelined. The control signal coordinates the system which is produced by the control unit. The random access memory component stores the 32-bit vector states and it can support two write and six read operations. The advantages of the architecture are high throughput achieves one random numbers per cycle as well as it does not require any off-chip memory.

4.2 Data Compression

Based on compression scheme method, linear type FSR reseeding can be improved by rearranging partitioning and merging test data from the given test data set, using fixed-length LFSR of maximum length sequence can be decompressed. During the LFSR run in autonomous mode, from the load pattern scan, LFSR seed is determined with the sequence

of bit generated matches the load pattern scan of specified bits. Due to the repeated patterns in the consecutive seeds the compression cannot be obtained, thus by eliminating repeated patterns the compression can be achieved. Since the on-chip decoder is not exploited, LFSR with the single polynomial is used, so that the decompression process is simple and fast. Initially n-bit LFSR is loaded with initial seed, applied by the automatic test equipment (ATE) as shown in Fig.15. In autonomous mode, LFSR utilizes the m cycles to fill the m-bit scan chain using the valid test vectors. The ATPG process pooled to overcome the drawback of LFSR-based compression.

Figure 15 : LFSR-reseeding Architecture

The constrained ATPG process problem are overcomes by the new LFSR-based compression [87] approach. For attaining a special purpose of slice, LFSR seed to encode as many slices as possible using the techniques, it specifies the last slice of each seed with the end of seedÂ's usage. To impose minimum compression overhead, inherent test characteristic set and stop slices is proposed using the stop-slice generator. This technique requires an additional hardware overhead for implementing the architecture to the general LFSR-based. The seed calculation algorithm is accompanied by the technique to reduce the number of seed calculations. Furthermost of the common techniques are the code based [88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99] and the linear decompressor based [100, 101, 102, 103, 104, 105] in the test-data compression techniques.

4.3 High-Speed Communication Cryptography

The high-speed and secure data-rate is decisive of fast keying algorithm, in the area of optic fibre based communication. Cryptographic key generation [106] using non-linear product feedback shift registers based approach to generate multiple key bits in each clock cycle for the digital communication on optic fibre which operates at several gigabits per second. For the indispensable high speed data transmission, the speed limitation cannot met by the LFSR circuit of sequential in nature. To overcome the speed limitation and to achieve the high-speed application of optical communication systems where several gigabits per seconds are required to achieve the necessary throughput using the parallel architecture.

NLFSR based approach utilizes n LFSRs which is used to select a new encoded majority function as shown in Fig.16, during each clock cycle with guaranteed asymptotic maximal length properties. For the output length of n = 64 and 31 input LFSRs, need 64-bit encoded majority functions ($f_0^M - f_{63}^M$).

Figure 16 : NLFSR-Encoded Majority Function Blocks

Each encoded majority function has three inputs, one from each LFSR. For implementing the number of unique encoded majority functions possible is $2^3C_4 = 70$, of which 64 are chosen for random. A secured demand for data communication increases NLFSR generates key of 64-bit in each clock cycle, is an effective and efficient approach towards fast key generation necessary for secure high data-rate communications.

4.4 Bose-Chaudhuri-Hocquenghem (BCH) Encoder and Cyclic Redundancy Check (CRC) Operation

The series or parallel shift register of type linear is an authoritative sequence element broadly classified and used in CRC operations and BCH encoders [5],[107, 108] are broadly used in the area of upcoming communication systems to detect the transmission errors in all communication protocols. The large scope of parallel architecture have been suggested in the literature of CRC operations and BCH encoders to large enough to accommodate the length of the transmitted increase the throughput [57]. In parallel CRC [109, 110] message. The bit generator and the first n output bits, implementations as shown in Fig.17 have been proposed $x(0), x(1), \cdots, x(n)$ computationally infeasible based on mathematical deduction. BCH encoders elucidate with high-speed architectures have been proposed [110, 111].

Figure 17 : CRC Architecture

Any generator polynomials be used for any LFSR, generates other bits forward or backward within the given division and multiplication computations. Consider a polynomial $g(x) = 1 + a + a^8 + a^9$ to generator the sequence. By using the formulation in equation (36) and equation (37) given by,

$$y(n) = y(n-9) + y(n-8) + y(n-1) + f(n) \tag{36}$$

where

$$f(n) = u(n-9) + u(n-8) + u(n-1) \tag{37}$$

Table 4 : Data Flow of CRC Architecture

Table 4 shows the flow of data at different time slots in the architecture. To compute the output, the architecture requires 17 clock cycles is depicted in the Table 4. After every clock cycles. feedback is reset to zero like basic LFSRs circuit. Since the input values are reset to zero, the registers are not affected as it retains the intermediate value for the computation, till the remainder bits are shifted out.

4.5 Stream Cipher Generator

Secured key stream generators cryptographically [112], the stream cipher as shown in Fig.18, clock period are large enough to accommodate the length of the transmitted message. Further, the dynamic key sequence are generated for the stream cipher and implemented for secured cryptographic transmission [113]. The bit generator and the first n output bits, $x(0), x(1), \cdots x(n-1)$, are computationally infeasible to predict the $(n+1)^{th}$ bit, $x(n)$ in the sequence with better than a half chance. The cryptanalyst will not generate other bits forward or backward within the given portion of the output sequence. The LFSRs operated

Figure 18 : Stream Cipher

at higher rates than the required output rate [114] with the multispeed generator exploits two LFSRs clocked at different speeds [63] as shown in Fig.19. The *LFSR-2* architecture, of degree n is clocked at a speed $s \geq 2$ times as fast as *LFSR-1* of degree l, $n \geq l$, and the output signal $d(t)$ as shown in equation (38) is produced according to

$$d(t) = \sum_{i=0}^{l-1} u(t+i)v(dt+i) \tag{38}$$

As a portion of the security key, variable d is used as the speed factor. If $LFSR-1$ and $LFSR-2$ have primitive feedback polynomials and $(l,n)=1$, $(s,2^n-1)=1$, then the output sequence would have a linear complexity with period $T=(2^l-1)(2^n-1)$ and possess exceptional statistical properties. Once the feedback polynomials are well known,

Figure 19 : Pseudo-Random Bit Generator Based on Massey-Ruppel's Multispeed Generator

the cryptanalyst can determine the speed factor s and the initial states of both LFSRs by $(s_{\max}-1)(2^l-1)$ consistency tests applied to an output section of length $N \geq l+n+\log_2 s_{\max}$ [26]. The hash values are determined by the LFSRs, which are used to generate the non- recursive pseudorandom sequence to encrypt and decrypt data stream in the stream cipher [115].

4.6 Test Pattern Generator

The system on chip testing and validation expects the hardware testing rapidity with low cost and high performance. In IC testing, a simple vector with an adequate number of patterns is generated to cover the high fault. The algorithmically generated test pattern generates good fault coverage for logic networks are extremely time-consuming for the sequential circuits, at which the test can be applied with the limitation of speed. Using linear feedback shift registers, pseudorandom test patterns are generated [59] is adopted as method for testing application [116, 117, 118], to achieve the low hardware overhead this testing method eliminates the necessity to store the deterministic test patterns. The non-linear feedback shift register used to generate non-recursive test pattern sequence is adopted for built-in self-test (BIST) [119]. The desired fault coverage is achieved with the number of test pattern by using LFSR as test pattern generator.

Pseudorandom patterns are generated using primitive LFSR is considered inadequate as a test pattern generator in many cases. Several research have been conducted to steer the LFSRs performance for random fault resistant, to increase the probability of generating test patterns. The test pattern generator as shown in Fig.20, consists of LFSRs with k stages of 2-port register $(D_1, D_2, D_3, \cdots, D_n)$, the feedback logic function with the XOR gates with the additional gates are distributed among the next stage registers. To generate the necessary seed values from the functional block, all the n bits of LFSR not necessarily to be inverted. To test the CUT completely, all the seeds generated using the m logic gates of XORs $(m < n)$.

Figure 20 : Test Pattern Generator

4.7 ***Image Encryption Using Stream Cipher***

The theory of information security is gaining more importance, in the field of data transmission and storage. Image processing techniques and medical imaging Images are broadly used for several applications in the area of images. Thus, it is important to protect the image data from the unapproved access. To hide the information, image encryption [120] plays a significant role in this field. Image encryption algorithms and methods as shown in Fig.21, which are dependable frequency domain and convoluted simple spatial domain. Initially, the encryption algorithms [121] are designed for the text image data, which is not secured due to huge data sizes for many real time multimedia applications. In commercial systems, to process the huge data of image and video, the software implementations of ciphers are usually too slow and hardware implementation substantially increases the cost of device manufacturers and service providers.

Figure 21 : Image Encryption Architecture

In the area of secure multimedia distribution, a major development to reduce the computational requirements by which dataÂ's are, encrypted partially using Â"selective encryptionÂ". For the security of digital image encryption, high and low levels of encryption standards are utilized. The visual quality of the encrypted images are degraded to that of the real image, in low-level encryption the image contents are visible and readable form to the viewers. In case of high-level security, the images look like random noise which is not readable to the viewers, the data contents are scrambled completely. To ensure a secured encryption, the selective encryption method is implemented to avoid encryption of all digital image bits. The lossless algorithm for image encryption and decryption [120], hence the images are of highly information are used in such applications and lose of information is not adequate. To obtain a fast method to encrypt only part of bit-stream [122, 123] is the key aspect in the image encryption.

5 Conclusion

This paper depicts the concrete amount of survey of feedback shift registers, either linear or non-linear product of fibonacci and galois configurations. It presents the most development of its sequences, algorithms, architectures and applications pertaining to analysing feedback functions of maximum period during last five decades. Thus, highlights the empirical approach to minimize the hardware gate complexity by choosing the states for the binary sequence to determine the next state function. For the past fifty years, quite a few architectures and algorithms are adopted to improve the rapidity of linear or non-linear type FSRs based stream cipher, due to the feedback functions applied to each and every bit substantially to reduces the depth of feedback function enactment. Thus, increasing the throughput efficiency with the reduction in the propagation time delay. Besides the various applications in diversified areas which include image encryption and decryption, cryptographic key stream generation, pseudo random bit generator, test pattern generator and communication apart from technical and scientific computations in general have been explored. Research on the parallel LFSR is deployed in high-speed optical communication systems, which requires several gigabits for the throughput. Due to the improvement in the

latency reduction and its throughput, FSRs probably fits for many real-time and high-speed applications.The latency-area-speed-accuracy are balanced for various heuristic algorithms and applications probably probed in detail and imposed within the scope of future work.

References

[1] N. Zierler. Several binary sequence generators. *Massachusetts Inst of Tech Lexington Lincoln Lab, Lexington, Massachusetts*, Sept. 1955.

[2] S. W. Golomb. Sequences with Randomness Properties. *Martin Co., Baltimore, Md.*, page p.62, 1955.

[3] D.A. Huffman. A linear circuit viewpoint on error-correcting codes. *IRE Trans. on, Information Theory*, 2(3):20–28, Sept. 1956.

[4] Bernard Elspas. The Theory of Autonomous Linear Sequential Networks. *IRE Trans. on Circuit Theory*, 6(1):45–60, Mar. 1959.

[5] W.W. Peterson and D.T. Brown. Cyclic Codes for Error Detection. *Proc. of the IRE*, 49(1):228–235, Jan. 1961.

[6] Franco P. Preparata. State-Logic Relations for Autonomous Sequential Networks. *IEEE Trans. on Electronic Computers*, EC-13(5):542–548, Oct. 1964.

[7] M.Y. Hsiao and K.Y. Sih. Serial-to-Parallel Transformation of Linear-Feedback Shift-Register Circuits. *IEEE Trans. on Electronic Computers*, EC-13(6):738–740, Dec. 1964.

[8] J. Massey and Ruey-Wen Liu. Equivalence of nonlinear shift-registers. *IEEE Trans. on Information Theory*, 10(4):378–379, Oct. 1964.

[9] P. A Bryant. Theory of feedback shift registers with invertors. *Proc. of the Institution of Electrical Engineers*, pages 1599–1605, 1969.

[10] W.A. Davis. Logical Design Using Shift Registers. *IEEE Trans. on Computers*, C-18(10):958–960, Oct. 1969.

[11] J.L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. on Information Theory*, 15(1):122–127, Jan. 1969.

[12] D. A Brown. Theory of feedback shift registers with invertors. *Proc. of the Institution of Electrical Engineers*, page 1434, 1970.

[13] D. A Green. Polynomial representation of nonlinear feedback shift registers. *Proc. of the Institution of Electrical Engineers*, pages 56–60, 1970.

[14] D. A Green. Nonlinear product-feedback shift registers. *Proc. of the Institution of Electrical Engineers*, pages 681–686, 1970.

[15] A. M. Patel. A multi-channel CRC register. *Proc. of the spring joint computer conference (AFIPS '71)*, 38:11–14, May 1971.

[16] E. Roth. Permitations arranged around a cycle. *Amer. Math. Monthly*, pages 990–992, 1971.

[17] E.J. Van Lantschoot. Double Adjacencies Between Cycles of a Circulating Shift Register. *IEEE Trans. on Computers*, 22(10):944–955, Oct. 1973.

[18] Harold Fredricksen. A class of nonlinear de Bruijn cycles. *Journal of Combinatorial Theory, Series A*, 19(2):192 – 199, 1975.

[19] Harold Fredricksen and Irving J. Kessler. Lexicographic Compositions and deBruijn Sequences. *J. Comb. Theory, Ser. A*, pages 17–30, 1977.

[20] H. M. Fredricksen and J. Maiorana. Neklaces of beads in k colors and k-ary de Bruijn sequences. *Discete Mathematics,North-Holland Publishing Company*, 23:207–210, 1978.

[21] J. Mykkeltveit. Generating and Counting the Double Adjacencies in a Pure Circulating Shift Register. *IEEE Trans. on Computers*, C-24(3):299–304, Mar. 1975.

[22] J. Mykkeltveit. Nonlinear recurrences and arithmetic codes. *Information and Control*, 33(3):193–209, 1977.

[23] A Ralston. A New Memoryless Algorithm for De Bruijn Sequences. *J. Algorithms.*, pages 50–62, 1981.

[24] H. Beker and F. Piper. *Cipher systems: the protection of communications*. New electronics communications international book. Northwood Books, 1982.

[25] E. McCluskey. Built-in self-test techniques. *IEEE Design and Test archive*, 2(3):21–28, Mar. 1985.

[26] K. Zeng, C. Yang, D. Wei, and T. R. N. Rao. Pseudorandom bit generators in stream-cipher cryptography. *IEEE Transactions on Computers*, 24(2):8–17, Feb. 1991.

[27] J. McCluskey. High speed calculation of cyclic redundancy codes. *In Proc. of ACM/SIGDA seventh international symposium on Field programmable gate arrays, New York, USA, ACM*, pages 250–256, 1999.

[28] J. Rajski G. Mrugalski and J. Tyszer. Ring generators - New devices for embedded test applications. *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(9):1306 – 1320, 2004.

[29] E. Dubrova. A Scalable Method for Constructing Galois NLFSRs With Period $2^n - 1$ Using Cross-Join Pairs. *IEEE Transactions on Information Theory*, 59(1):703–709, Jan. 2013.

[30] B. Schneier. Applied Cryptography. *John Wiley and Sons,Inc.*, 1996.

[31] E. Zenner. Built-In Test for VLSI -Pseudorandom Techniques, Reihe Informatik; No. 04-2004. *Cryptanalysis of LFSR-based Pseudorandom Generators - a Survey, University of Mannheim*, page 28, 1987.

[32] A. Canteaut. Open problems related to algebraic attacks on stream ciphers. *Proc. of international conference on Coding and Cryptography, WCC '05,Bergen, Norway*, pages 120–134, 2005.

[33] S.Golomb. Shift Register Sequences. *Aegean Park Press. 1982*.

[34] Nan Li and Elena Dubrova. An Algorithm for Constructing a Smallest Register with Non-Linear Update Generating a Given Sequence. *in Proceedings of IEEE International Symposium on Multiple-Valued Logic (ISMVL'2014)*, 19-21 May 2014.

[35] H. Fredricksen. A survey of full length nonlinear shift registers cycle algorithms. *Society for Industrial and Mathematics Review.*, 24(2):195–221, Apr. 1982.

[36] C. J. Jansen. Investigations On Nonlinear Streamcipher Systems: Construction and Evaluation Methods. *Ph.D. Thesis, Technical University of Delft*, 1989.

[37] E.Dubrova. A List of Maximum-Period NLFSRs. *Cryptology ePrint Archive. Report 2012/166, http://eprint.iacr.org/2012/166*, Mar. 2012.

[38] E. Dubrova. An Equivalence Preserving Transformation from the Fibonacci to the Galois NLFSRs. *ArXive ePrint Technical Report 0801.4079, arXiv: 0801.4079*, 2008.

[39] A. Fredricksen, H. MGill. Disjoint Cycles From The De Bruijn Graph. *University of Southern California Los Angeles Electronic Sciences Lab*, 10(52):1599–1605, June 1968.

[40] F.S. Annexstein. Generating De Bruijn sequences: an efficient implementation. *IEEE Trans. on Computers*, 46(2):198–200, Feb. 1997.

[41] Taejoo Chang, Bongjoo Park, Yun Hee Kim, and Iickho Song. An efficient implementation of the D-homomorphism for generation of de Bruijn sequences. *IEEE Trans. on Information Theory*, 45(4):1280–1283, May 1999.

[42] I. S. T. Chang. Some properties of cross-join pairs in maximum length linear sequences. *In Proc. of ISZTA*, pages 1077–1079, 1990.

[43] M. T. E.Dubrova. Analysis and Synthesis of (n,k)-Non-Linear Feedback Shift Registers. *Proc. of Design, Automation and Test in Europe (DATE '2008),Munich, Germany*, pages 1286–1291, 10-14 Mar. 2008.

[44] T. Etzion and A. Lempel. Algorithms for the generation of full-length shift- register sequences. *IEEE Trans. on Information Theory*, 30(3):480–484, May 1984.

[45] T. Helleseth and T. Klove. The number of cross-join pairs in maximum length linear sequences. *IEEE Trans. on Information Theory*, 37(6):1731–1733, Nov. 1991.

[46] I. Janicka-Lipska and J. Stoklosa. Boolean feedback functions for full-length nonlinear shift registers. *Telecommunications and Informatioin Technology*, 5(1):28–29, Apr. 2004.

[47] A. Klapper and M. Goresky. 2-adic Shift Registers. *Proc. Cambridge Security Workshop on Fast Software Encryption, Springer-Verlag*, pages 174–178, Apr. 1994.

[48] A. Klapper and M. Goresky. 2-adic Shift Registers. *Technical Report,Department of Computer Science, University of Kentucky*, 5(1):239–293, Apr. 1994.

[49] A. Klapper. Feedback with Carry Shift Registers over Finite Fields. *K.U. Leuven Workshop on Cryptographic Algorithms, Springer-Verlag*, 1008:170–178, 1995.

[50] A. Klapper and M. Goresky. Large Period Nearly de Bruijn FCSR Sequences. *Advances in Cryptology-EUROCRYPT '95 Proc., Springer-Verlag*, pages 263–273, 1995.

[51] A. Klapper and M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *J. Crypt. 10.*, pages 111–147, 1997.

[52] S. Mansouri J.-M. Chabloz and E. Dubrova. An algorithm for constructing a fastest Galois NLFSR generating a given sequence. *In C. Carlet and A. Pott, editors, Sequences and Their Applications-(SETA), Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, 6338(1):41–54, 2010.

[53] E. Dubrova. A Transformation From the Fibonacci to the Galois NLFSRs. *IEEE Trans. on Information Theory*, 55(11):5263–5271, Nov. 2009.

[54] E. Dubrova. Finding Matching Initial States for Equivalent NLFSRs in the Fibonacci and the Galois Configurations. *IEEE Trans. on Information Theory*, 56(6):2961–2966, June 2010.

[55] Vladimir Raphael Rosenfeld. Enumerating De Bruijn sequences. *MATCH Communications in Mathematical and in Computer Chemistry*, 45:71–83, 2002.

[56] M. Ayinala and K.K. Parhi. High-Speed Parallel Architectures for Linear Feedback Shift Registers. *IEEE Trans. on Signal Processing*, 59(9):4459–4469, Sept. 2011.

[57] J. Massey. Review of Theory and Practice of Error Control Codes (Blahut, R.E.; 1983). *IEEE Transactions on Information Theory*, 31(4):553–554, July 1985.

[58] P. Koopman and T. Chakravarty. Cyclic redundancy code (CRC) polynomial selection for embedded networks. *International Conference on Dependable Systems and Networks*, pages 145–154, June 2004.

[59] W. H. McAnney P. H. Bardell and J. Savir. Built-In Test for VLSI -Pseudorandom Techniques. *John Wiley and Sons, Inc.*, 1987.

[60] Erin Casey. Berlekamp-Massey Algorithm. *University of Minnesota, REU Summer*, 2000.

[61] P. Udaya. Euclid's algorithm and LFSR synthesis. *Proceedings IEEE International Symposium on Information Theory*, page 420, 25-30 June 2000.

[62] J. H. Derby. High speed CRC computation using state-space transformation. *Proc. Global Telecommun. Conf. (GLOBECOM'01)*, 1(2):166–170, 2001.

[63] J.L. Massey and R.A. Rueppel. Linear Ciphers and Random-sequence Generators with Multiple Clocks. *Proc. Eurocrypt '84, Springer-Verlag Lecture Notes in Computer Science,New York*, 209:74–87, 1985.

[64] Elena Dubrova. A Method for Generating Full Cycles by a Composition of NLFSRs. *Design, Codes and Cryptography, Springer*, pages 822–836, 2014.

[65] E. Dubrova. Synthesis of Binary Machines. *IEEE Trans. on Information Theory*, 57(10):6890–6893, Oct. 2011.

[66] A. Klapper. A Survey of Feedback with Carry Shift Registers. *Sequences and Their Applications - SETA 2004, LNCS*, 3486(1):56–710, 2004.

[67] M. Goresky and A.M. Klapper. Fibonacci and Galois representations of feedback-with-carry shift registers. *IEEE Trans. on Information Theory*, 48(11):2826–2836, Nov. 2004.

[68] J. Noras. Fast pseudorandom sequence generators: Linear feedback shift registers, cellular automata, and carry feedback shift registers. *Univ. Bradford Elec. Eng. Dept., Bradford, U.K.*, Rep. 94, 1997.

[69] A. Klapper. Distributional properties of d-FCSR sequences. *Journal of Complexity*, 20(2-3):305–317, 2004.

[70] Martin Voros. Algebraic Attack on Stream Ciphers. *Master's thesis, submitted to COMENIUS UNIVERSITY, Department of Computer Science*, 2007.

[71] A. A. Bruen and R. A. Mollin. Cryptography and Shift Registers. *The Open Mathematics Journal*, pages 16–21, 2009.

[72] J. Pelzl C. Paar. Understanding Cryptography, Chapter 2- stream Cipher. *Springer-Verlag, Berlin Heidelberg*, pages 29–54, 2010.

[73] Erik Zenner Gregor Leander and Philip Hawkes. Cache Timing Analysis of LFSR-based Stream Ciphers. *Proc. Crypto and Coding,Springer LNCS*, 2:5921, 2009.

[74] D.T. Tang and Chin-Long Chen. Logic Test Pattern Generation Using Linear Codes. *IEEE Trans. on Computers*, C-33(9):845–850, Sept. 1984.

[75] F. Arnault, T. Berger, M. Minier, and B. Pousse. Revisiting LFSRs for Cryptographic Applications. *IEEE Transactions on Information Theory*, 57(12):8095–8113, Dec. 2011.

[76] T. Matsumoto, M.; Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. on Modeling and Computer Simulation.*, 8 (1), Jan. 1998.

[77] Mariko Hagita Mutsuo Saito Makoto Matsumoto, Takuji Nishimura. Mersenne Twister and Fubuki Stream/Block Cipher. *Cryptology ePrint Archive, Report 2005/165*, June 2005.

[78] Shrutisagar Chandrasekaran and Abbes Amira. High Performance FPGA implementation of the Mersenne Twister. *4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA)*, pages 482–485, Jan. 2008.

[79] G. Marsaglia and A. Zaman. A new class of random number generators. *Annals of Applied Probability.*, 1(3):462–480, Mar. 1991.

[80] L'Ecuyer P. Tezuka, S. and R. Couture. On the lattice structure of add-with-carry and subtract-with-borrow random number generators. *ACM Trans. Model. Comput. Simul.*, 3:315 – 331, Nov. 1995.

[81] R. Coveyou and R. Macpherson. Fourier analysis of uniform random number generators. *J. ACM.*, 14(Issue 1):100–119, Jan. 1967.

[82] D. Knuth. The Art of Computer Programming. *Seminumerical Algorithms, 3rd edition. Addison Wesley, Reading Mass*, 2, 1997.

[83] G. Marsaglia. Yet another rng. *Proc. stat. math. Posted to electronic bulletin board sci.*, Aug. 1994.

[84] A. Klapper and M Goresky. Feedback shift registers, combiners with memory, and arithmetic codes. *Department of Computer Science, University of Kentucky, Tech.*, page Rep. No. 239 : 93, 1993.

[85] George Marsaglia. Xorshift RNGs. *Journal of Statistical Software.*, 8(Issue 14), July 2003.

[86] Jiang Jiang Yuan Li, Paul Chow and Minxuan Zhang. Software/Hardware Framework for Generating Parallel Long-Period Random Numbers Using the WELL Method. *21st International Conference on Field Programmable Logic and Application*, pages 110–115, 5-7 Sept. 2011.

[87] M. Koutsoupia, E. Kalligeros, X. Kavousianos, and D. Nikolos. LFSR-based test data compression with self-stoppable seeds. *Design, Automation Test in Europe Conference Exhibition.*, pages 1482–1487, 2009.

[88] A. Chandra and K Chakrabarty. Test data compression and decompression based on internal scan chains and Golomb coding. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 21(6):715–722, 2002.

[89] A. Chandra and K Chakrabarty. A unified approach to reduce SOC test data volume, scan power and testing time. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(3):352–363, 2003.

[90] A. Chandra and K Chakrabarty. Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes. *IEEE Trans. on Computers*, 52(8):1076–1088, Aug. 2003.

[91] P.T. Gonciari, B.M. Al-Hashimi, and N. Nicolici. Variable-length input Huffman coding for system-on-a-chip test. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):783–796, June 2003.

[92] A. Jas, J. Ghosh-Dastidar, Mom-Eng Ng, and N.A. Touba. An efficient test vector compression scheme using selective Huffman coding. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):797–806, 2003.

[93] X. Kavousianos, E. Kalligeros, and D. Nikolos. Test Data Compression Based on Variable-to-Variable Huffman Encoding With Codeword Reusability. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1333–1338, 2008.

[94] X. Kavousianos, E. Kalligeros, and D. Nikolos. Multilevel-Huffman Test-Data Compression for IP Cores With Multiple Scan Chains. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 16(7):926–931, 2008.

[95] A.H. El-Maleh and R.H. Al-Abaji. Extended frequency-directed run-length code with improved application to system-on-a-chip test data compression. 2:449–452, 2002.

[96] M. Nourani and M. H. Tehranipour. RL-Huffman encoding for test compression and power reduction in scan applications. *ACM Trans. Des. Autom. of Electr. Syst.*, 10:91–115, Jan. 2005.

[97] S. Reda and A. Orailoglu. Reducing test application time through test data mutation encoding. *Proc. of Design, Automation and Test in Europe Conference and Exhibition*, pages 387–393, 2002.

[98] P. Rosinger et al. Simultaneous reduction in volume of test data and power dissipation for systems-on-a-chip. *Electronics Letters*, 37(24):1434–1436, 2001.

[99] M Tehranipoor, M. Nourani, and K Chakrabarty. Nine-coded compression technique for testing embedded cores in SoCs. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 13(6):719–731, 2005.

[100] B. Koenemann. LFSR-coded test patterns for scan design. *Proc.in IEEE European Test Conference*, pages 237–242, 1991.

[101] A. Jas C. V. Krishna and N. A. Touba. Test vector encoding using partial LFSR reseeding. *Proc. International Test Conference*, pages 885–893, Nov. 2001.

[102] C. V. Krishna and N. A. Touba. Reducing test data volume using LFSR reseeding with seed compression. *Proc. International Test Conference*, pages 321–330, 2002.

[103] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee. Embedded deterministic test. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(5):776–792, 2004.

[104] X. Kavousianos V. Tenentes and E. Kalligeros. State skip LFSRs: bridging the gap between test data compression and test set embedding for IP cores. *Design, Automation and Test in Europe, 2008.*, pages 474–479, Mar. 2008.

[105] E. Volkerink and S. Mitra. Efficient seed utilization for reseeding based compression. *VLSI Test Symposium, 2003. Proc. 21st*, pages 232–237, May 2003.

[106] Bradley Johnson Sunil P. Khatri Pey-Chang Kent Lin, Alex Ivanov. A novel cryptographic key exchange scheme using resistors. *IEEE 29th International Conference on Computer Design (ICCD)*, pages 451–452, Oct. 2011.

[107] T.-B. Pei and C. Zukowski. High-speed parallel CRC circuits in VLSI. *IEEE Trans. on Communications*, 40(4):653–657, Apr. 1992.

[108] T.V. Ramabadran and S.S. Gaitonde. A tutorial on CRC computations. *IEEE Micro*, 8(4):62–75, Aug. 1988.

[109] K.K. Parhi. Eliminating the fanout bottleneck in parallel long BCH encoders. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 51(3):512–516, Mar. 2004.

[110] Xinmiao Zhang and K.K. Parhi. High-speed architectures for parallel long BCH encoders. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 13(7):872–877, July 2005.

[111] A. Doering. Concepts and Experiments for Optimizing Wide-Input Streaming CRC Circuits. *23rd International Conference on architecture of Computing Systems (ARCS)*, pages 1–5, 22-23 Feb. 2010.

[112] P.-C. K. Khatri. VLSI Implementation of a Non-Linear Feedback Shift Register for High-Speed Cryptography Applications. *GLSVLSI '10,Providence, Rhode Island, USA: ACM 978-1-4503-0012-4/10/06.*, May 2010.

[113] S. Pal, K.K. Soundra Pandian, and K.C. Ray. FPGA implementation of stream cipher using Toeplitz Hash function. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1834–1838, Sept. 2014.

[114] R.A. Rueppel. When Shift Registers Clock Themselves. *Proc. Eurocrypt '87, Springer- Verlag Lecture Notes in Computer Science,New York*, 304:53–64, 1987.

[115] V.S. Lakshmi P.P. Deepthi, P.S. Sathidevi. A new hardware efficient stream cipher based on hash functions. *International Journal of Communication Networks and Distributed Systems (IJCNDS)*, 3(4):340–361, 2009.

[116] J. A. Waicukauski E. B. Eichelberger, E. Lindbloom and T. W. Williams. Structured Logic Testing. *Englewood Cliffs, NJ: PrenticeHall*, 1991.

[117] P. Agrawal and V.D. Agrawal. Probabilistic Analysis of Random Test Generation Method for Irredundant Combinational Logic Networks. *IEEE Transactions on Computers*, C-24(7):691–695, July 1975.

[118] J. Savir, Gary S. Ditlow, and Paul H. Bardell. Random Pattern Testability. *IEEE Transactions on Computers*, C-33(1):79–90, Jan. 1984.

[119] Jon C. Muzio Dandan Qi. Non-linear test pattern generators for built-in self-test. *International Journal of Communication Networks and Distributed Systems (IJCNDS)*, 1(2):179–194, 2008.

[120] S. B. Lala Krikor. Image Encryption Using DCT and Stream Cipher. *European Journal of Scientific Research.*, 32(1):48–58, 2009.

[121] Tannishtha Som. Subhra Mazumdar. Data Encryption with Linear Feedback Shift Register. *International Journal of Scientific and Engineering*, 3,(Issue 6), June 2012.

[122] Xiliang Liu. Selective encryption of multimedia content in distribution networks: challenges and new directions. *Proc. of Communications, Internet, and Information Technology (CIIT 2003), Scottsdale, AZ, USA*, pages 48–58, Nov. 2003.

[123] Babel M. Fonteneau C., Motsch J. and D ´eforges O. A hierarchical selective encryption technique in a scalable image codec. *International Conference in Communications, Bucharest, Romania*, pages 1–4, June 2008.

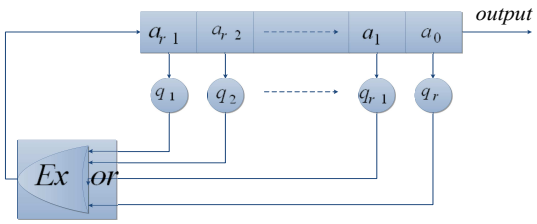

Figure 1: Fibonacci LFSR

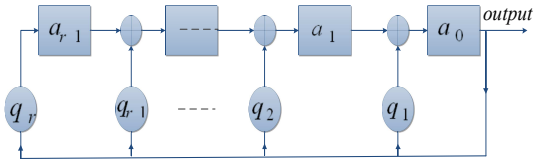

Figure 2: Galois LFSR

	Fibonacci LFSR *Simple shift register generator* (SSRG)	**Galois LFSR** *Multiple-return shift register generator* (MRSRG) or *Modular shift register generator* (MSRG)
Definition	External XORs as well as many-to-one LFSR	Internal XORs as well as one-to-many LFSR
Modulo-2 summation	Weighted Modulo-2 summation blocks from the feed forward path	Weighted Modulo-2 summations in the feed forward path
Switching Speed	Fibonacci is slower than galois due to the additional number of hardware logic gates in the feedback path	Galois form is generally faster than the Fibonacci in hardware due to the reduced number of logic gates in the feedback loop
Feedback Taps	The set of feedback taps for the equivalent Fibonacci generator is denoted as $[f_1,\ m - f_2, m - f_3, \cdots, m - f_n]f$	The set of feedback taps for a Galois generator is denoted as $[f_1,\ f_2, f_3, \cdots, f_n]g$

Table 1: Comparison of Fibonacci and Galois LFSR

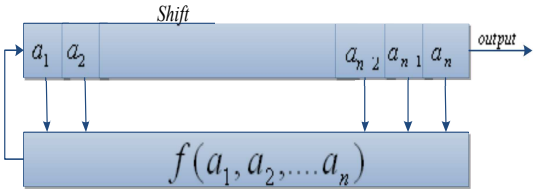

Figure 3: General Structure of FSR

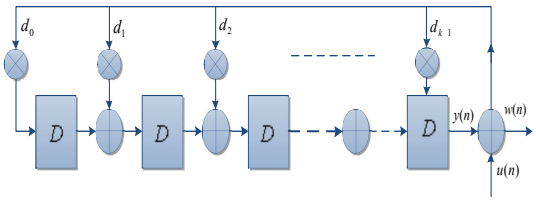

Figure 4: General LFSR Architecture

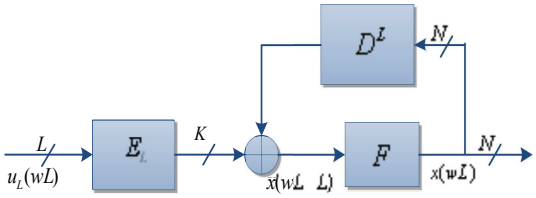

Figure 5: Parallel LFSR Architecture

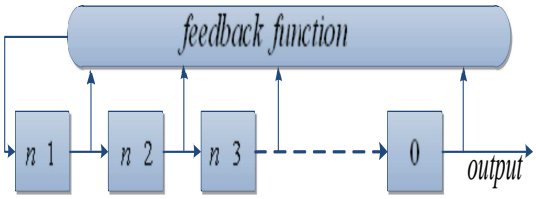

Figure 6: An n-bit Fibonacci NLFSR Architecture

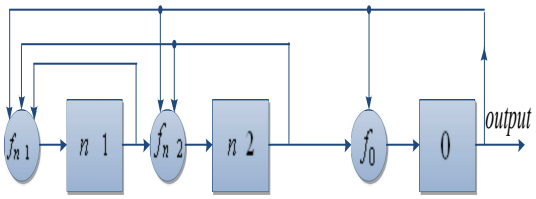

Figure 7: An n-bit Galois NLFSR Architecture

	Fibonacci NLFSR $N_F(n-bit)$	**Galois NLFSR** $N_G\ (n-bit)$
Feedback Function	Feedback is applied to the $n-1\ th$ bits $f'_{n-1} = a_0 \oplus g_{n-1} \oplus g_{n-2}\mid_{+2} \oplus \cdots g_\tau \mid_{+n-1-\tau}$	Feedback is applied to every bits $f_{n-1} = a_0 \oplus g_{n-1}$ $f_{n-2} = a_{n-1} \oplus g_{n-2}$
Propagation Time	Comparatively higher than galois	Reduced since the bits are computed in parallel
1^{st} and 2^{nd} golomb postulate	Satisfy both postulates	Don't satisfy both postulates
Period of Output sequence	Equal to the longest cyclic sequence of its consecutive states	Not equal to the longest cyclic sequence of its consecutive states

Table 2: Comparison of Fibonacci and Galois NLFSR

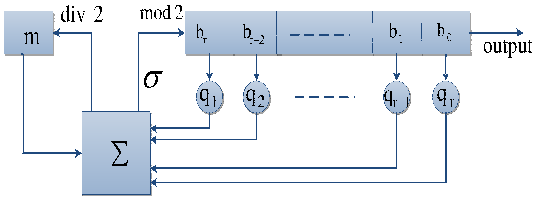

Figure 8: Fibonacci FCSR Architecture

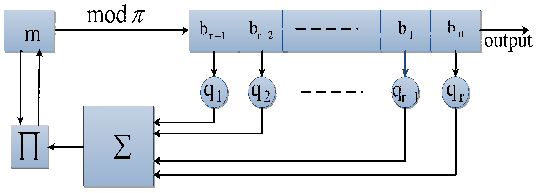

Figure 9: Fibonacci d-FCSR Architecture

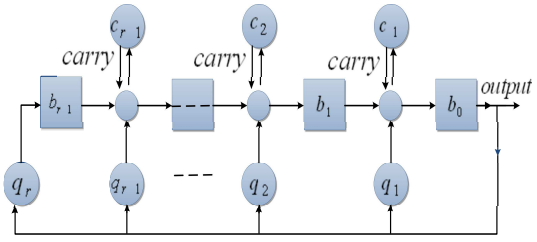

Figure 10: Galois FCSR Architecture

Figure 11 : Galois d-FCSR Architecture

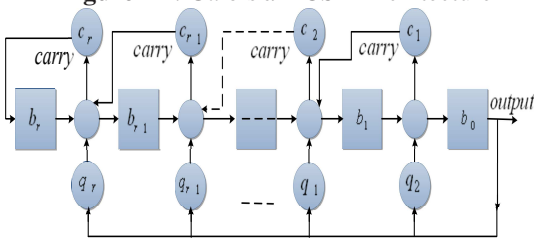

Figure 11: Galois d-FCSR Architecture

Parameter	**LFSR**	**FCSR**
Mathematical Model	LFSRs is rational series in the ring $GF(2)[[x]]$	FCSRs is provided by rational 2-adic numbers
Transition Function	LFSR is linear	FCSR is quadratic
Attacks	Weak resistance to correlation and algebraic attacks	Resistance to correlation and algebraic attacks
Register Flip-Flops	n	n
Carry Flip-Flops	0	$m-1$
XOR Gates	$m-1$	$4(m-1)$
AND Gates	0	$m-1$

Table 3: Comparison of LFSR and FCSR

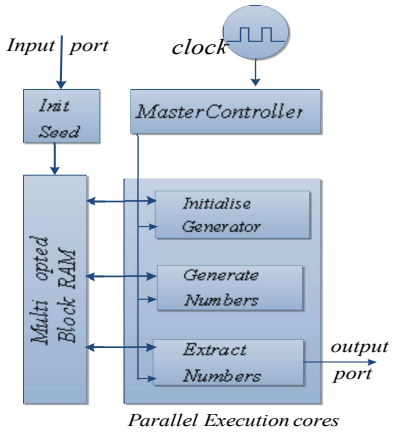

Figure 12: MT19937 Hardware Architecture

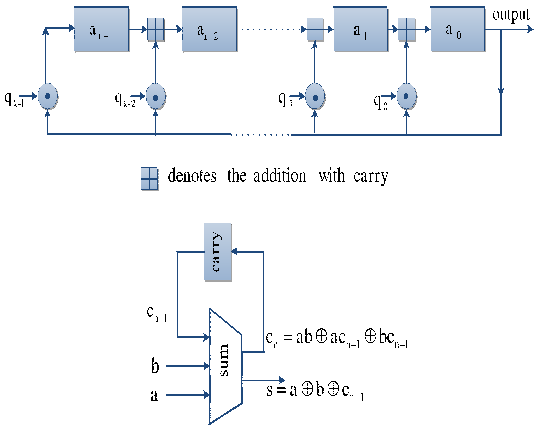

Figure 13: Add-With-Carry Generator

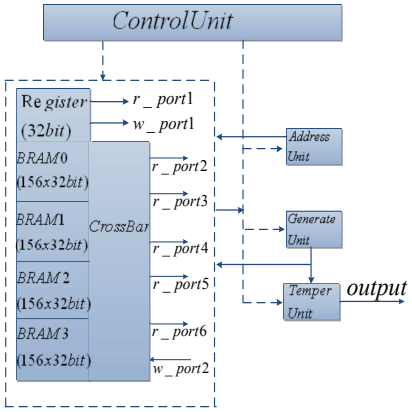

Figure 14: WELL Hardware Architecture

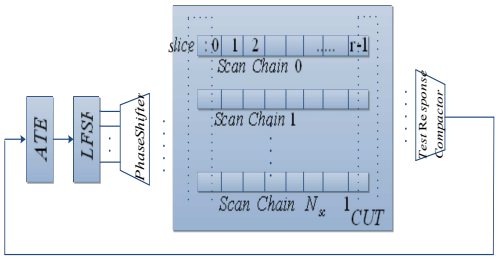

Figure 15: LFSR-reseeding Architecture

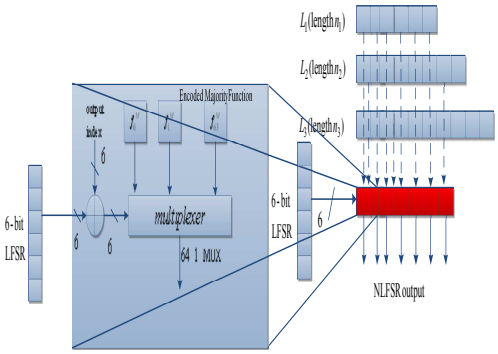

Figure 16: NLFSR-Encoded Majority Function Blocks

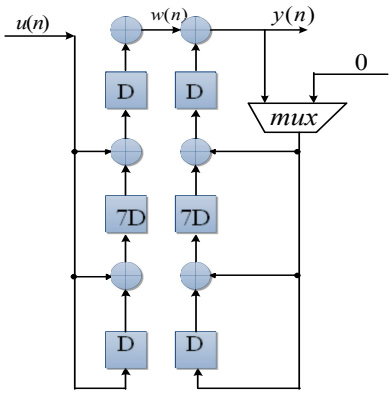

Figure 17: CRC Architecture

clock	u(n)	f(n)	y(n)	clock	u(n)	f(n)	y(n)
0	0	0	0	**9**	0	1	0
1	1	1	1	**10**	0	1	1
2	0	0	1	**11**	0	1	0
3	1	1	0	**12**	0	1	1
4	0	0	0	**13**	0	0	1
5	1	1	1	**14**	0	1	0
6	1	1	0	**15**	0	1	1
7	0	0	0	**16**	0	1	1
8	1	0	0	**17**	0	0	0

Table 4: Data Flow of CRC Architecture

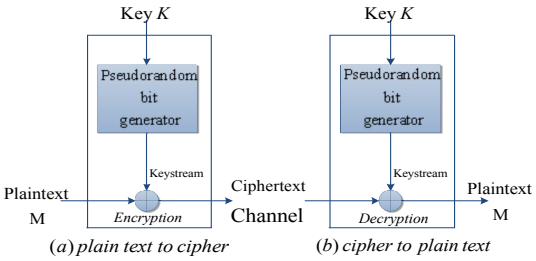

Figure 18: Stream Cipher

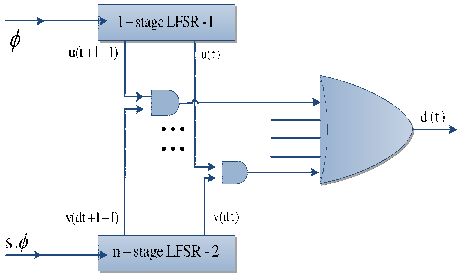

Figure 19: Pseudo-Random Bit Generator Based on Massey-Ruppel's Multispeed Generator

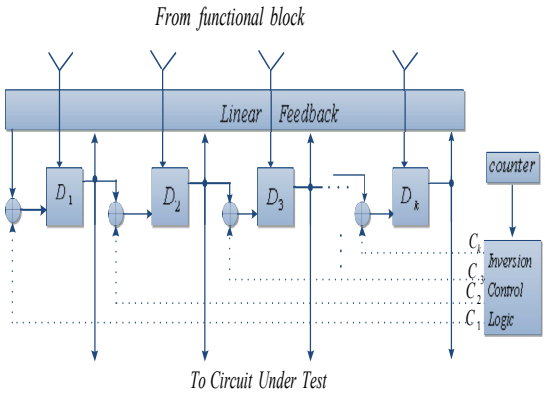

Figure 20: Test Pattern Generator

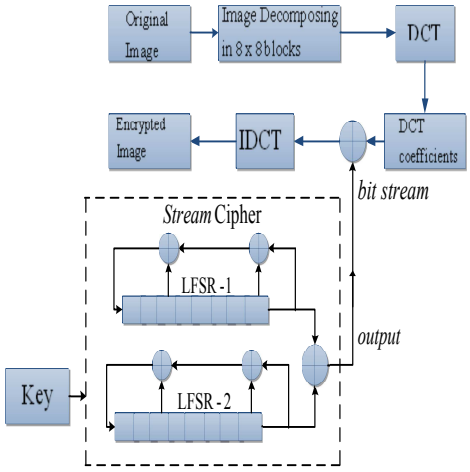

Figure 21: Image Encryption Architecture