© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Design of a 1.9 GHz low-power LFSR Circuit using the Reed-Solomon Algorithm for Pseudo-Random Test Pattern Generation

Vishnupriya Shivakumar[1*], C. Senthilpari, Zubaida Yusoff[1]

[1]Faculty of Engineering,
Multimedia University, 63100, Jalan Multimedia, Cyberjaya, Selangor, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2021.13.06.020
Received 16 July 2020; Accepted 30 May 2021; Available online 31 August 2021

Abstract: A linear feedback shift register (LFSR) has been frequently used in the Built-in Self-Test (BIST) designs for the pseudo-random test pattern generation. The higher volume of the test patterns and the lower test power consumption are the key features in the large complex designs. The motivation of this study is to generate efficient pseudo-random test patterns by the proposed LFSR and to be applied in the BIST designs. For the BIST designs, the proposed LFSR satisfied with the main strategies such as re-seeding and lesser test power consumption. However, the reseeding approach was utilized by the maximum-length pseudo-random test patterns. The objective of this paper is to propose a new LFSR circuit based on the proposed Reed-Solomon (RS) algorithm. The RS algorithm is created by considering the factors of the maximum length test patterns with a minimum distance over the time *t*. Also, it has been achieved an effective generation of test patterns over a stage of complexity order $O(m \log_2 m)$, where *m* denotes the total number of message bits. We analysed our RS LFSR mathematically using the feedback polynomial function to decrease the area overhead occupied in the designs. The simulation works of the proposed RS LFSR bit-wise stages are simulated using the TSMC *130 nm* on the Mentor Graphics IC design platform. Experimental results showed that the proposed LFSR achieved the effective pseudo-random test patterns with a lower test power consumption of *25.13 µW* and *49.9 µs*. In addition, proposed LFSR along with existing authors' LFSR are applied in the BIST design to examine their power consumption. Ultimately, overall simulations operated with the highest operating frequency environment as *1.9 GHz*.

Keywords: Reed-Solomon (RS), Galois Field (GF), Linear Feedback Shift Register (LFSR), Test Pattern Generator (TPG), Built-in Self-Test (BIST)

1. Introduction

Recently, many designers are actively focused on the low power improvements in VLSI technology. Linear feedback shift registers (LFSRs) are the most frequently used method for generating the test patterns. Conventional test pattern generators (TPGs) have high switching power due to an increase in the transition delay, which is not suitable for many types of applications, especially battery-operated devices. Hence, an LFSR is favorable because it can generate random test patterns with a small percentage of switching power [1]. Other factors such as low power consumption, performance, and the silicon area consumption of a random test pattern generator are also important. An LFSR is used to generate pseudo-random test patterns since it has excellent properties, such as a lower silicon area overhead in the chip [2]. In general, an LFSR consists of a series of flip-flops to perform the shift register function. A bit pattern enables the feedback of the shift registers. The shift register rotates the series of input message bits by shifting to the left or the right, and the shifted bits are fed back into the shift register using an exclusive OR operation [3]. The original value of the shift register is said to be the initial value. Depending on this initial value, the shift register generates test patterns [4].

**Vishnupriya Shivakumar: vishnu.priya120@gmail.com*

A TPG is used in many applications such as circuit design, cryptography, and Monte Carlo methods, the latter of which require high quality and efficient test patterns [5]. To obtain an efficient implementation of the test pattern, it satisfies some properties. Firstly, the test patterns should be reproducible; secondly, the more extended length test patterns should be generated over a long period; and finally, the storage for random test patterns should be used efficiently in a circuit. Computational researchers need to be careful while selecting a TPG algorithm since it may fail for randomness [6].

An LFSR can be implemented using the Reed-Solomon (RS) codes for the generation of test patterns. The RS code is a type of error-correcting code initially proposed by Irving S. Reed and Gustave Solomon in 1960 and was later modified by other authors for their specific needs. These codes can detect and correct multiple-bit errors in transmitted output test patterns [7]. Moreover, an RS code is a type of cyclic code, meaning that the binary symbols' 1' and '0' can be complemented along the transmission line. During the transmission of data bits, the RS codes are synchronized in terms of false recognition. An analytical notation has been developed for the message bits to reduce false recognition in the RS codes [8]. In modern systems, the input seeds are encoded and distributed along the transmission line. If the failure occurs at a node, it should be replaced efficiently using the remaining functional nodes [9].

The RS codes have been mathematically analyzed using the Galois field. The Galois field used the finite field arithmetic, named by French algebraist Galois [10]. These fields are constructed using finite numbers, especially prime numbers, and are represented using an algebraic notation [11]. The RS codes are effectively used in cryptographic applications for error detection and correction [12]. The Galois-field-based RS codes can be calculated using either of two methods: the first uses a software programming algorithm, and the second uses a hardware design [13]. RS codes are also called the regenerating and forward error-correcting codes. It means that the RS codes can identify the faulty nodes and recover them. It is used in many storage system applications such as CDs, DVDs, etc. Besides, it is widely used in bar codes, QR codes, digital video broadcasting, and wireless satellite communications [6], [14].

In this paper, the proposed LFSR circuit and its auxiliary circuits are designed and simulated using a Mentor Graphics TSMC 130 nm tool. Related literature works of the LFSR design are discussed in Section 2. Section 3 briefed the modified algebraic design of RS codes. It achieved the maximum length patterns within the minimum distances. Also, the mathematical analysis of the proposed RS LFSR was calculated in terms of the feedback polynomial function. In Section 4, the RS algorithm and the RS LFSR circuit are proposed. Finally, the proposed LFSR was designed using the CMOS logic technique, and the design has been verified in the IC station. Section 5 discussed the results obtained for the proposed circuit and its auxiliary components. The proposed LFSR circuit was compared with the existing designs of LFSRs and tabulated. Additionally, the proposed LFSR is applied in the BIST design to identify its less power consumption with the existing authors' LFSR.

2. Related Works of the LFSR

Sian et al. [15] proposed using RS codes as an iterative structure and presented an algebraic notation for the input seeds. According to Swastik et al. [16], the RS codes can be defined using two fields: GF(q) and GF(q^m); however, in this paper, GF(q^m) was chosen to give a better transmission. For example, the RS codes can transmit 128 random bits for a seven-bit input message sequence in the finite field 'F'. According to Amin et al. [17], the distance of the message bits transmitted is denoted by the factor d, which is the divisor of any multiplicative group in the Galois field. Han et al. [18] show that the RS codes are linear error-correcting codes that play a vital role in many applications. The message bits are transmitted with the minimum distance. If we suppose that the received random test pattern is dissimilar from the expected pattern, it is identified as a fault, also known as an erasure. Erasures need to be detected and corrected by the parity bits to enable better transmission. In this paper, the XOR gates are used for a parity checker. Oscar et al. [19] presented an LFSR circuit that contains a series of flip-flops and the XOR gate as a switch. As per [20], the length of LFSR is assumed as L in this paper; the initial stage of the LFSR is referred to as a seed.

Meanwhile, the positions of the bit change in successive stages are referred to as tapping. Depending on the tap value, the values of the output polynomial sequences change. According to Jia-Min et al. [21], the LFSR should generate pseudo-random test patterns parallel for all the k subsequent stages simultaneously over various input seeds.

Perenzoni et al. [22] proposed the TPG for the image sensor, which is being used in a spacecraft. It is also implemented on a digital silicon-on-chip with higher power consumption. Niclass et al. [23] designed the linear arrays of the LFSR counter. A five-stage pipelined structure in the counter is designed and implemented in a 180 nm CMOS design. The performance values need improvement for these proposed designs. Bronzi et al. [24] designed an array of the counter using a 9-bit LFSR. Rather, the reduced delay in the circuit can be improved with low noise levels. The author mentioned as an extension of the proposed design can be applied in molecular imaging. Pavia et al. [25] proposed the LFSR design as a counter used in the SRAM memory. Since the LFSR design is designed based on the CMOS differential buffer structure and the same clock distribution, the propagation delay in the designs gets increased. Daniel et al. [26] used the LFSR circuit as a counter to reduce the decoding logic in large-scale array applications. The authors' decoding logic of the LFSR can also be implemented algorithmically to motivate high-end security. Considering the existing LFSR designs, the proposed method of LFSR should be achieved with less power consumption and maximum length patterns.

3. Mathematical Analysis and Design of the Proposed RS LFSR

The RS code for (*L*, *m*) is used for transmission, where *L* is the block length of the LFSR, *m* is the input message bit or initial seed value. The parity bits (*2t*) to be added to correct the faulty bits during the transmission [27]. We choose the block length of the message bits as L= pm −1, where p is a prime number for the RS codes. The field properties of the RS codes are studied in terms of the Galois field. A group of the Galois field is defined as (*F*, + (or) *, *1*), where *F* denotes a set of elements in the field, '+' denoted an adder operation, '*' denoted a multiplier operation, and $1 \in F$ is identity [6].

3.1 Modified RS Algebraic Design

Consider a series of input message bits $m_0, m_1,\ldots, m_{k-1}$ that is mapped to the polynomial for the Galois field GF(p^m), where *p* is a prime number and there exist *k* polynomial estimations such that $m(x_0) = z_0$, $m(x_1) = z_1,\ldots, m(x_{k-1}) = z_{k-1}$. Using the Lagrange method [6], the message bits polynomial can be written as in equation (1),

$$m(x) = \sum_{i=0}^{m-1} z_a \prod_{j=0}^{m-1} \frac{x-x_j}{x_i-x_j} \tag{1}$$

Each error code for the random test pattern needs to be corrected with the parity bits. It is, therefore, necessary to consider a polynomial function for the parity bits. Let *2t* be the number of parity bits for the message *m*(*x*), for *k* coefficients and *2t* distinct non-zero points $x_0, x_1,\ldots, x_{2t-1}$. The symbol for the parity polynomial is assumed to be P, i.e. P= *2t*, which is expanded in equation (2) as,

$$\begin{gathered} m_0 + m_1 \,.\, x_0 + \ldots\ldots\ldots\ldots. + m_{k-1} \,.\, x_0^{k-1} = P_0 \\ m_0 + m_1 \,.\, x_1 + \ldots\ldots\ldots\ldots. + m_{k-1} \,.\, x_1^{k-1} = P_1 \\ \cdot \\ \cdot \\ m_0 + m_1 \,.\, x_{k-1} + \ldots\ldots\ldots\ldots. + m_{k-1} \,.\, x_{k-1}^{k-1} = P_{k-1} \end{gathered} \tag{2}$$

Since the RS codes are cyclic, the coefficients of the message bits *m*(*x*) exist only in a finite field *GF*(p^m). The generator polynomial g(*x*) for the finite field is shown in equation (3) as,

$$g(x) = \prod_{j=1}^{L-m} (x - \alpha^j) \tag{3}$$

For example, the generator polynomial for the four-bit test patterns can be written as,

$$g(x) = x^4 - \alpha^3\, x^3 + x^2 - \alpha\, x + \alpha^3 \tag{4}$$

The above equation (4), relates the individual test pattern bits to the generator polynomial coefficients in terms of *x*, and α^1, α^2,….., α^{2t-1} , which are the distinct non-zero message bits in the Galois field *F*.
The generator polynomial *g*(*x*) can also be defined for the true case as $p(X) = g(x).q(x) \equiv 0 \; (mod \;\; x^L -1)$, where *p*(*X*) is the remainder and *q*(*x*) is the quotient for the modulo-2 operation. The modulo-2 operation can be carried out using the XOR operation.

$$\sum_{i=0}^{k-1} U_i \alpha_{ij} X_{ij} = 0 \qquad \text{where } i, j = 1,2,\ldots,m \tag{5}$$

The outputs of classical RS codes are assumed to be $U(x) = (p_x(a_1) \ldots p_x(a_n))$, where p_x represents the parity bits for the correction. *U* is a fixed integer that lies between zero and *k*. The final RS codes output are then represented by equation (5), as the product of the RS codes output function and the input message bits.

3.2 Minimum Distances Achieved for the Maximum Length Pseudo-Random Test Patterns

The outputs could be generated the maximum length in the pseudo-random patterns by considering the modified RS codes. The maximum length pseudo-random patterns with the minimum distances are examined using the concepts of multiplicity in the codes. The multiplicity structure in the codes should be identified using the multivariant low-degree polynomial and the restriction of the univariant low-degree polynomial [16]. Therefore, the polynomial for the LFSR is fixed as low-degree multivariant bits. Since the RS codes are linear, the minimum distance is assumed as $d \geq (L-m+1)$. It can be minimized further, and the distance can be rewritten in *d+1* [7]. It can be seen that the binary code parameters are in the set of $\{L=p^m, d+1\}$. Here, *p* is the number of prime values, *m* is the message bit, and *d+1* is the minimum distance. For example, a sequence code for the message bits of *15,11,7,5,1,* (for *p=16*), the minimum distances achieved are *2,4,6,8,16*, respectively. Consequently, the minimum distances as *d+1*, the maximum length as *m*, pseudo-random bit patterns as *p*, with the total block length as *L* are identified in Table-1.

Table 1 - The RS LFSR achieved with minimum distance for various message bits

L=p^{m}											
p=2		**p=4**		**p=8**		**p=16**		**p=32**		**p=64**	
m	**d+1**	**m**	**d+1**	**m**	**d+1**	**m**	**d+1**	**m**	**d+1**	**m**	**d+1**
1	2	3	2	7	2	15	2	31	2	63	2
		1	4	4	4	11	4	26	4	57	4
				1	8	7	6	21	6	51	6
						5	8	16	8	45	8
						1	16	11	12	39	10
								6	16	36	12
								1	32	30	14
										24	16
										18	22
										16	24
										10	28
										7	32
										1	64

3.3 Analysis of the Proposed RS LFSR Test Patterns using the Feedback Polynomial Function

The proposed RS LFSR consists of *m*-bit shift registers; the XOR gate and the multiplexers act as the Galois adder and multiplier. The concepts of the Galois adder and the multiplier are introduced to re-seed the maximum length test patterns. The feedback polynomial expressions are prescribed for the area-sensitive designs.

In general [5], the LFSR feedback polynomial of degree *m* is given by,

$$F_m(X) = \alpha_0 X_0 + \alpha_1 X_1 + \dots\dots\dots\dots + \alpha_m X_m \tag{6}$$

Equation (5) is the feedback polynomial function, which needs to satisfy certain properties such as, $\alpha_0 = \alpha_m = 1,\ \alpha_i = \{0,1\}$, and $\alpha_1 = \alpha_2 = \cdots = \alpha_{k-1} = 0$, where α is the message bits. The polynomial chosen should exhibit two specific conditions for the maximum length of the pseudo-random test patterns: firstly, it should be a primitive polynomial; and secondly, for any smaller values of $k = 2^m - 1$, the subjective polynomial can be divided by the polynomial $1+X^k$, where *k* is any number between zero and *m*. Hence, the feedback polynomial for the proposed RS LFSR *m*-bit can be written in equation (7) as,

$$F_m(X) = 1 + \alpha_0 X_0 + \alpha_1 X_1 + \alpha_2 X_2 + \dots + \alpha_m X_m \tag{7}$$

The proposed test patterns are mapped in the vector notation. The initial stage of the feedback is represented as $\overrightarrow{Y(0)}$ in equation (8),

$$\overrightarrow{Y(0)} = [y_1^{(0)} \quad y_2^{(0)} \ \dots\dots\dots y_m^{(0)}\ ]^{\mathrm{T}} \tag{8}$$

Hence, the prediction of the successive stages are characterized as, $\overrightarrow{Y(1)} = \mathrm{I} \oplus \overrightarrow{Y(0)}$. Finally, the k^{th} stage of the LFSR output can be calculated according to the expression below.

$$\overrightarrow{Y(K)} = \mathrm{I}^k \oplus \overrightarrow{Y(0)}, \qquad \text{where } \mathrm{I}^k = \mathrm{I} \oplus \mathrm{I}^{k-1} \tag{9}$$

$$\text{where, } I = \begin{bmatrix} 0 & 0 & \cdots & 0 & \alpha_0 \\ 1 & 0 & \cdots & 0 & \alpha_1 \\ 0 & 1 & \cdots & 0 & \alpha_2 \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & \alpha_m \end{bmatrix}$$

By using equation (6), the four-bit stage of the proposed RS LFSR can be briefed as,

$$\mathrm{F}_4(\mathrm{X}) = 1 + \alpha_0 X^0 + \alpha_1 X^1 + \alpha_2 X^2 + \alpha_3 X^3 \tag{10}$$

In the above equation (10), $F_4(X)$ is denoted as the feedback polynomial function for a four-bit stage. Hence, the initial feedback of the proposed LFSR can be mapped in vector notation as below.

$$\overrightarrow{Y(0)} = [y_1^{(0)} \quad y_2^{(0)} \quad y_3^{(0)} \quad y_4^{(0)}\ ]^{T} \tag{11}$$

Using equation (8), the subsequent two feedback stages of the proposed LFSR written below.

$$\overrightarrow{Y(1)} = I \oplus \overrightarrow{Y(0)} = \begin{bmatrix}0&0&0&1\\1&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix} \oplus \begin{bmatrix}y_1^{(0)}\\y_2^{(0)}\\y_3^{(0)}\\y_4^{(0)}\end{bmatrix} = \begin{bmatrix}y_4^{(0)}\\y_1^{(0)}\\y_2^{(0)}\\y_3^{(0)}\end{bmatrix} = \begin{bmatrix}y_1^{(1)}\\y_2^{(1)}\\y_3^{(1)}\\y_4^{(1)}\end{bmatrix} \tag{12}$$

$$\overrightarrow{Y(2)} = I^2 \oplus \overrightarrow{Y(0)} = \begin{bmatrix}0&0&0&1\\1&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix}^2 \oplus \begin{bmatrix}y_1^{(0)}\\y_2^{(0)}\\y_3^{(0)}\\y_4^{(0)}\end{bmatrix} = \begin{bmatrix}y_3^{(0)}\\y_4^{(0)}\\y_1^{(0)}\\y_2^{(0)}\end{bmatrix} = \begin{bmatrix}y_1^{(2)}\\y_2^{(2)}\\y_3^{(2)}\\y_4^{(2)}\end{bmatrix} \tag{13}$$

Based on equations (10), (11), and (12), the four-bit stage of the proposed LFSR test patterns generated as 1100, 0110, 0011, 1001. The successive patterns of the LFSR are generated simultaneously by using the bit-wise XOR operator.

$$\overrightarrow{Y(0)} = [1 \quad 1 \quad 0 \quad 0]^{T} \tag{14}$$

$$\overrightarrow{Y(1)} = [0 \quad 1 \quad 1 \quad 0]^{T} \tag{15}$$

$$\overrightarrow{Y(2)} = [0 \quad 0 \quad 1 \quad 1]^{T} \tag{16}$$

$$\overrightarrow{Y(3)} = [1 \quad 0 \quad 0 \quad 1]^{T} \tag{17}$$

The RS LFSR should be executed parallel for all k stages with the minimum distance as $L + i.k \geq d_{min}$. Here, L is the block length or the size of the LFSR, i is the number of successive stages of a shift register, and $d_{min} = d+1$ as the minimum distance achieved.

4. Proposed RS Algorithm and RS LFSR Circuit

The initial seed value of the RS LFSR is represented in terms of the message bits m in algorithm 1. The transmission of the message bits mapped onto a polynomial bit sequence p_x is ready for transmission into the shift registers. Simultaneously, the Galois field addition and multiplication operations are carried out to generate the multi-valued logic. The multi-valued logic is used for re-seeding the test patterns if required. Since the parity bits are added to the output for correction, it is calculated by the modulo operation. Followed by the successive iterations of the algorithmic steps, the RS LFSR output can be written as $1 + p(X) + X^{L-m}\,\alpha\,(x)$, where "1" is denoted as a primitive polynomial bit. The primitive bit was added to the initial polynomial for the proposed LFSR.

Algorithm 1: RS LFSR

```
Step 1: Input seed bits $m$
Step 2: Assign
          $m$ belongs to GF ($2^m$)
          Map $m$ bits to polynomial, $p_x(\alpha_i) = x_i$, i ∈ {1, 2, .... α}
          Block length $L = 2^m - 1\ \forall\ m>0$;
          Parity bits $2t = (L-m)$
          Generate polynomial bits $g(X) = X^L + 1$
Step 3: Procedure for $x$= 0 to $L$, do generate RS ($L$, $m$) ∃
          $X^{L-m}\,\alpha\,(x)$ is $[(q(x).\ g(x)) \oplus p(X)]$
          $p(X)$ is $[X^{L-m}\, m\,(x)\ \%\ g(x)]$
Step 4: Output of the RS LFSR as pseudorandom test patterns,
          $Y(X) = 1 + p(X) + X^{L-m}\,\alpha\,(x)$
Step 5: end
```

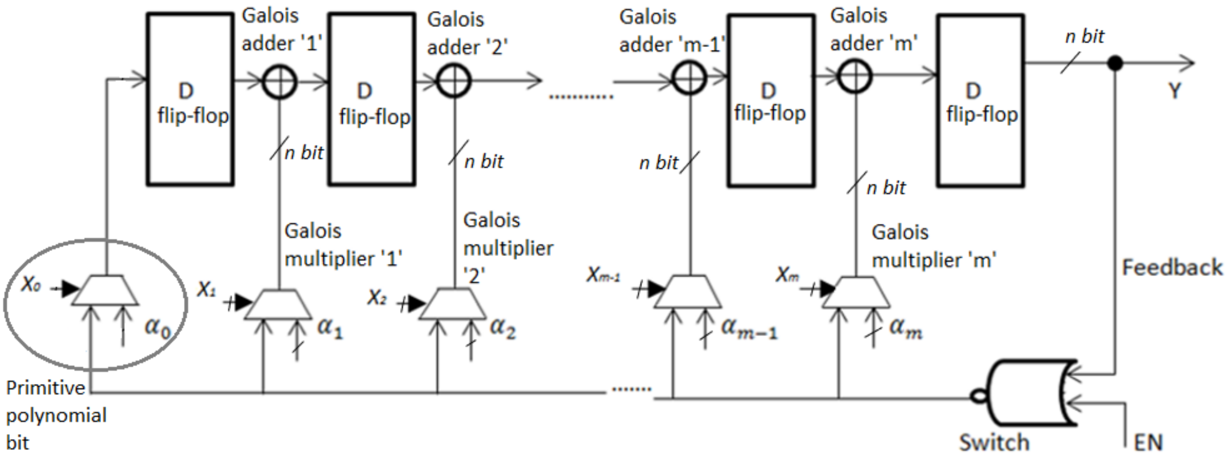


Fig. 1 - Block diagram of the proposed n-bit RS LFSR

Table 2 - Comparison table of proposed LFSR with the conventional LFSR

Method	Feedback polynomial function		#Test patterns	
	Conventional LFSR	Proposed LFSR	Conventional LFSR	Proposed LFSR
4-bit	X^4+X+1	$1+\alpha_0X_0+\alpha_1X_1+\ldots+\alpha_{15}X_{15}$	4	15
8-bit	$X^8+X^6+X^5+X+1$	$1+\alpha_0X_0+\alpha_1X_1+\ldots+\alpha_{255}X_{225}$	8	225
16-bit	$X^{16}+X^5+X^3+X^2+1$	$1+\alpha_0X_0+\ldots+\alpha_{65535}X_{65535}$	16	65535
m-bit	X^m+X+1	$1+\alpha_0X_0+\ldots+\alpha2^{m-1}\ X2^{m-1}$	m	2^{m-1}

The proposed RS LFSR test patterns and their feedback functions are listed with the conventional LFSR in Table 2. It identified as the proposed method could generate the maximum length patterns using their operation of the Galois field. The Galois operation for the adder and multiplier in the LFSR enlarged into the maximum length pseudo-random patterns in the LFSR by the concepts of subsets in their calculation. The subsets of the initial seed bits could be written as $\alpha_0 X^0$, $\alpha_1 X^1$, $\alpha_2 X^2$, …, $\alpha_n X^n$. In the proposed method, the last-bit of the feedback polynomial is defined as $\alpha 2^{m-1}\ X2^{m-1}$. Hence, the Galois field extended to the factor 2^{m-1} in the initial seeds, which generates the additional bits in LFSR than the conventional method.

An algorithm-based RS LFSR block is implemented in Fig.1 block diagram. In this block diagram, an XOR gate is used as a switch to enable (EN) the LFSR circuit. The LFSR continuously generates pseudo-random test patterns as long as the EN input is high. The series of the D flip-flops are used as shift registers to rotate the pseudo-random test patterns. The Galois adder is assumed to be the XOR gates and the Galois multiplier as a multiplier for their convenient circuit design. The primitive polynomial bit is used to generate the maximum length pseudo-random patterns [4]. Hence, the primitive polynomial bit for the RS LFSR is identified as one of the Galois multipliers fed back into the shift registers. The Galois adder and multiplier are designed up to *"m"* stages for generating the multi-valued logic. The multi-valued logic is used for re-seeding the pseudo-random test patterns to the *n*-bit values in the BIST designs. The multi-valued logic is also much beneficial in large-scale array applications.

It shows that the proposed RS LFSR can generate the maximum length of the test patterns. Although in the worst-case scenario, the proposed LFSR can generate the patterns until $2^{\frac{m}{2}-1}$ which is greater than the conventional LFSR. The conventional LFSR has the selected roots for pattern generation. Whereas in the proposed LFSR, the roots of the polynomial can be selected at any tapping value. Consequently, for the proposed *m*-bit multistage LFSR, the test patterns are constant. The stage that is needed for the proposed LFSR is $m \log_2 m$. Thus, the size of the proposed LFSR should be scaled proportionally to $m \log_2 m$ (or) $\frac{m}{2} - 1 \log_2 m$. The remaining stages of the proposed LFSR can be designed using the conventional CMOS logic techniques in the range of $2 \leq m \leq 10$.

As per the block diagram, the proposed LFSR simulated and verified for the four-stage random patterns. Fig. 2 shows a schematic diagram of an XOR gate, a multiplexer, and a D flip-flop constructed using the CMOS design style. These auxiliary circuits are used to design the bit-wise stage of the proposed LFSR. The XOR gate (a) transistors are sized in terms of the lesser critical path, giving lower power consumption. In the NAND based D flip-flop design (b), the inputs feed through the A and B and also by feed through factor Q and $\bar{Q}$ respectively due to its cross-coupled design. The logical effort of the NAND flash rising edge D flip-flop produces the output as a logical 'high' pattern. The design of the 2:1 multiplexer (c) in the schematic used the transmission gate (push-pull method) due to reducing the bit transition delay. The transmission gates are equally sized for PMOS and NMOS to maintain the restoring logic. In the XOR gate circuit, the logical level swing restoration is achieved using M_1, M_2, M_3, and M_4 transistors. The swing restoration in the D flip-flop and multiplexer is achieved by using M_1 and M_2 transistors.

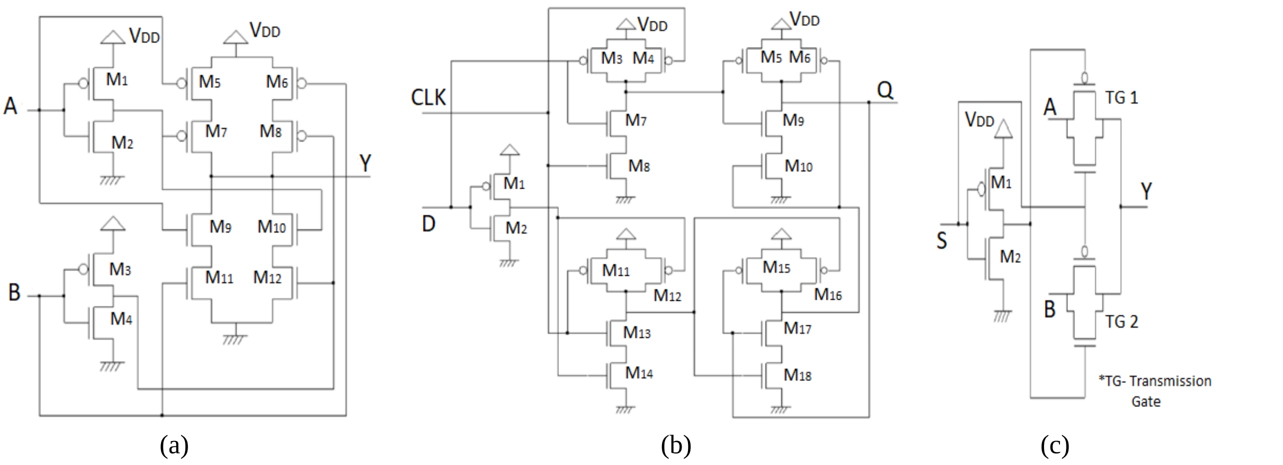


Fig. 2 - Transistor-level schematic of an (a) XOR gate; (b) D flip-flop; (c) multiplexer

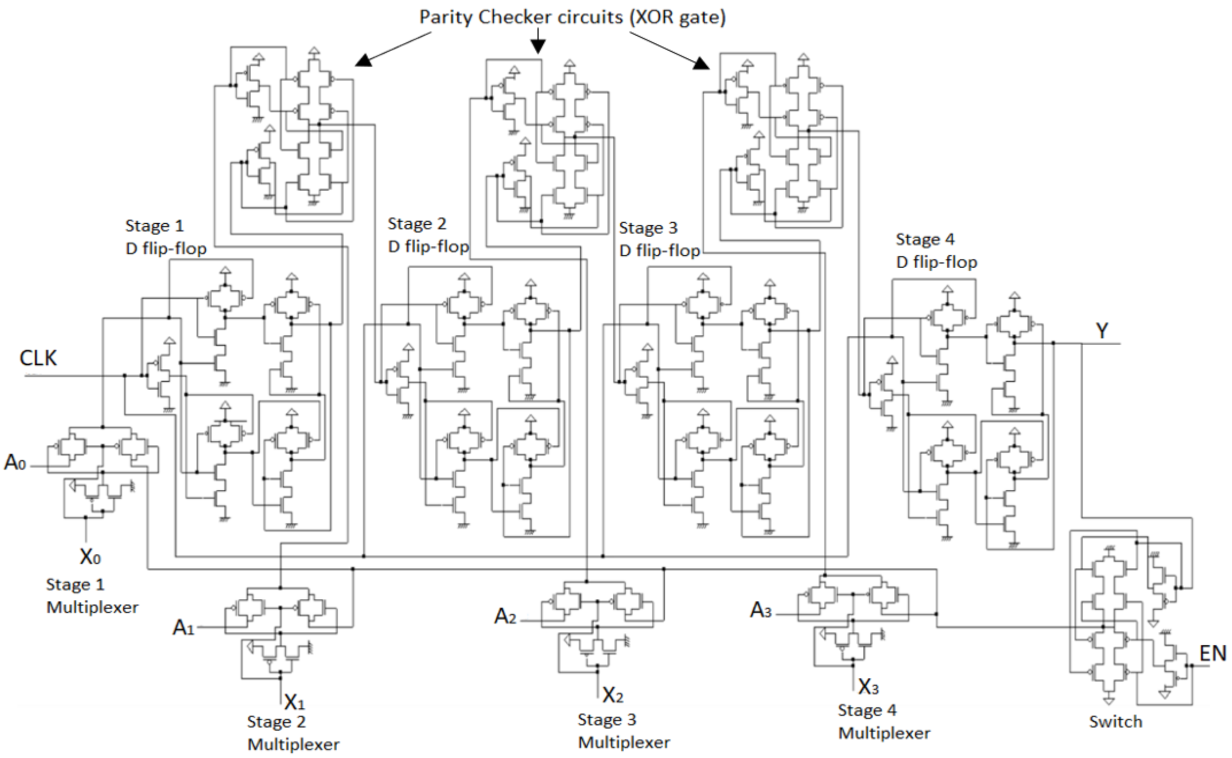


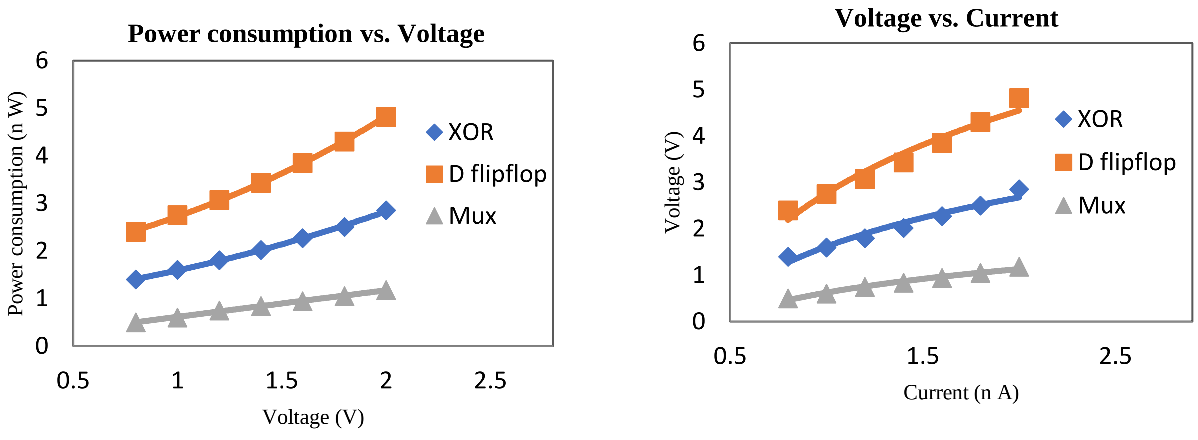
Fig. 3 - Transistor-level schematic of the proposed RS LFSR bit-wise scheme

Fig. 3 represents the schematic of the proposed LFSR circuit constructed using the auxiliary circuits. Most existing methods for the LFSR arrangement shown in the literature [1]–[3], [5] are the software (coded) model. In this paper, the hardware model of the LFSR arrangement is proposed. The proposed LFSR circuit used the multiplexer for bit-wise pattern selection. A_1, A_2, A_3, A_4 indicates the input seeds of the LFSR, and multiplexer select bits are assigned according to the enable input. Whereas X_1, X_2, X_3, X_4 are the generator polynomial bits assigned by the algorithmic values. The CMOS XOR gate is designed by an enhancement PMOS and NMOS transistor, which gives an equal bit transition in the output. As per the XOR operation, either of the one input is enabled, the output would be logically 'high,' which denotes the primitive bit. The primitive bit of LFSR feeds into a D flip-flop which makes one consecutive delay per bit. Normally XOR gates are used as a parity checker, which checks the input bit pattern that is enabled as either logical '1' (or) not. If no bit sequences are enabled, then the logic of the sequential patterns will not be recognized. Hence the D flip-flop consecutive delay and XOR checker must be an important factor for the regular bit-wise pattern. The stages of the multiplexer circuit used for bit-wise selection. The proposed four-bit stage LFSR is designed based on the dynamic logic technique. The cascaded auxiliary circuit restores the logical level at the output *Y* using the CMOS inverters. The different W/L factor sizing of the transistors maintained the effective resistance values while there is a significant increase in the parasitic capacitances.

5. Simulation Results and Discussions

The proposed RS LFSR circuit, simulated and verified using the Mentor Graphics IC design tool. The CMOS design of all the circuits is analyzed for the low power dissipation in the TSMC 130nm tool. The simulation results for the power and the current values over the different voltages for the auxiliary circuits are plotted in Fig. 4.

The CMOS XOR design used six pMOS and nMOS transistors. It dissipated the power as 1.4 nW. Due to the regular arrangement factors of the transistors, there is a minimized critical path with lesser power dissipation. The D flip-flop in the design consumes only 3 nW power, which is very small, and this is due to the standard logical effort of the cell. The multiplexer circuit has been designed using the transmission gates, which is nullifying the critical path. The outcome of the above-mentioned method dissipates the power of 0.1 nW and the delay of 0.14 ns. As shown in the block diagram of Fig. 2, all the auxiliary circuits are arranged regularly for the proposed LFSR circuit. It reduced the critical path and circuit delay. Overall, the LFSR circuit consumed the power as 25.13 µW and the delay as 49.9 ns. A standard voltage of 1.2 V is used for the critical analysis, and it operated at a frequency of 1.9 GHz. The overall performance values are shown in Table 3.



(a) (b)

Fig. 4 - Plots of Power consumption vs. Voltage and Voltage vs. Current for the auxiliary circuits

Table 3 - Performance factors of the proposed circuit and its auxiliary circuits

Performance	Feature size (nm)	Voltage (V)	Power (nW)		Delay (µs)		Frequency (GHz)	Area (µm^2)
			Pre-simulation	Post-simulation	Pre-simulation	Post-simulation		
XOR	130	1.2	0.0014	0.0029	0.72	0.9	1.9	13 x 11
D flipflop	130	1.2	0.003	0.0045	1.13	1.25	1.9	21 x 15
Mux	130	1.2	0.0001	0.00013	0.14	0.16	1.9	7 x 9
LFSR	130	1.2	25.13	34.8	49.9	56.7	1.9	133 x 30

The performance factors of the proposed LFSR with its auxiliary circuits are listed in Table 3. The pre-simulation and the post-simulation values of power and delay ate measured separately to identify its efficiency in the IC designs. The overall circuits could be operated in the 1.9 GHz frequencies with the less supply voltage of 1.2 V, which is well suited in the BIST operations. Also, the area occupied are measured using layout measurement of the particular drawn designs. The total circuit designs are enveloped with the low-level leakage transistors and the less transitions XOR gates. The lesser power consumption is achieved in the designs using eliminating the power leakage between the transistors. It could be accomplished by the principle of the dynamic-mode rather than the static-mode of operation.

The transient response experimented for the proposed LFSR, where the synchronous clock distribution for the D flip-flops is maintained logically high. The multiplexer transition stages are considered as a logical toggle for the output values. During the cascaded switching transitions of the multiplexer from logical low (t_{pLH}) to high (t_{pHL}), the output leads to numerous impulses highlighted in Fig 5. Moreover, the input pulse bits are assigned as the successive three pulse width delay values of 100 ns, 50 ns, and 25 ns to maintain high output V(Y). More signal impulses are eliminated in the responses using the weak transistors as an inverter in the circuits. Additionally, used fewer aspect ratio-ed transistors for the overall design to reduce the area overhead in the IC design.

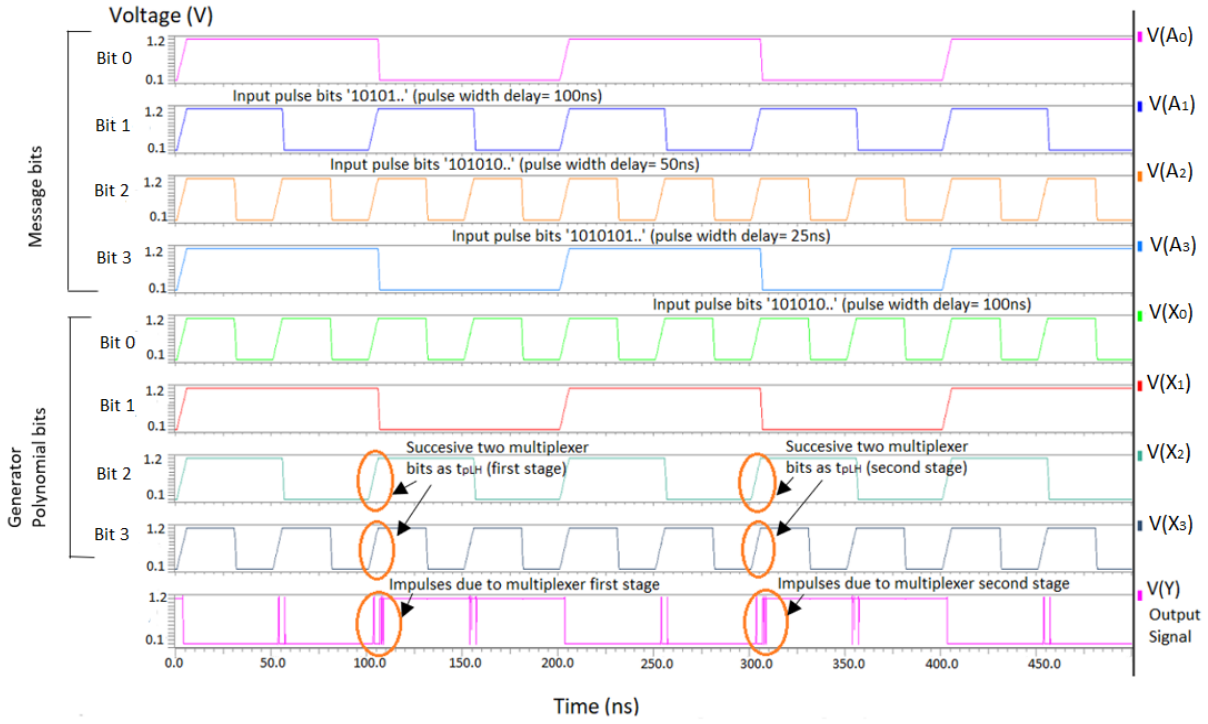


Fig. 5 - Transient response analysis of proposed RS LFSR scheme

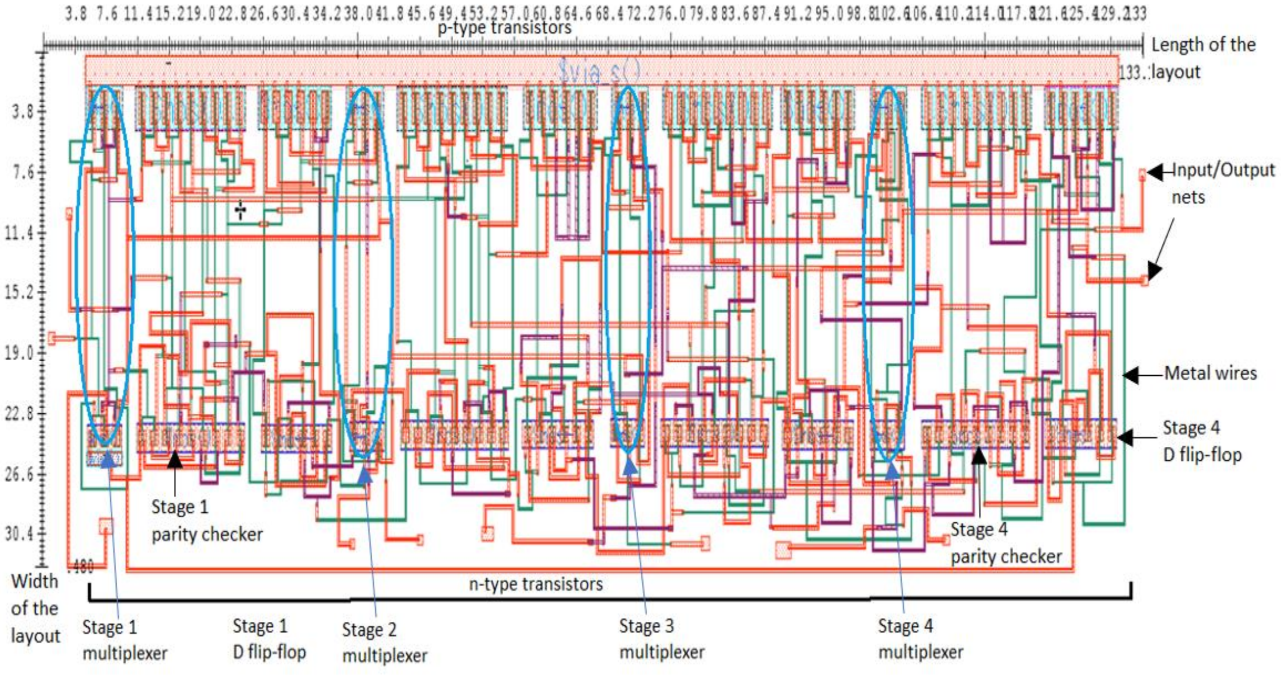
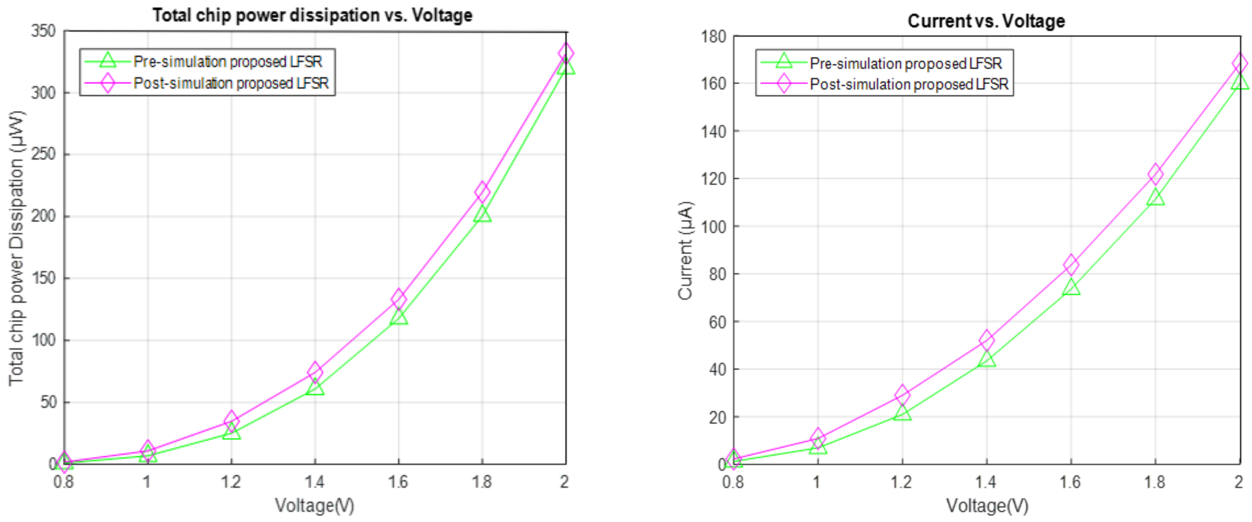


Fig. 6 - Schematic-Layout of the proposed bit-wise RS LFSR circuit design



(a) (b)

Fig. 7 - Pre-simulation and Post-simulation results of the proposed LFSR in the IC station. (a)Total chip power dissipation vs. Voltage; (b) Current vs. Voltage

Table 4 - Comparison of the performance of the proposed RS LFSR scheme with other simulation work by various authors

Author	Feature size (nm)	Transistor count	Voltage (V)	Power consumption	% Improvement	Delay (μs)	% Improvement	Frequency (GHz)	Area (μ m^2)	Power-delay product (μm^2-μs)	Area-delay product (fJ)
Perenzoni et al. [22]	130	148	1.2	47.7 mW	49.81	104	52.95	1.78	36001.4	496080	3744145.6
Niclass et al. [23]	130	152	1.2	530 mW	23.78	146.5	27.54	1.8	19362.5	776451	2836606.2
Bronzi et al. [24]	130	166	1.2	50 mW	51.20	124.6	52.84	1.814	7885.2	623055	982495.9
Pavia et.al. [25]	130	148	1.2	52 μW	89.62	79.9	64.25	1.25	4600.03	4154.8	367542.3
Daniel et al. [26]	130	136	1.2	157.5 μW	84.04	147.9	66.26	0.6	6446.1	23294.2	953378.1
Proposed RS LFSR	130	144	1.2	25.13 μW	--	49.9	--	1.9	3990	1253.9	199101

The layout of the proposed LFSR is drawn in Fig. 6. The layout is drawn at the schematic transistor level, and each transistor block is segmented based on the auxiliary circuits. The total area of the layout is measured as length x width of the drawn design occupied in the IC design station. The layout used substantial stray capacitance, wire capacitance, junction capacitance, and input-output long metal wire connections. The redundancy of the metal wires and the area can be constrained furthermore. For an area-sensitive design strategy, the proposed RS LFSR can be designed using the feedback polynomial function discussed mathematically in Section 3.3. However, the bit-slice methodology of the proposed LFSR is used to generate the TPG with the bit-wise test patterns. The bit-wise test patterns can be isolated from the output of the respective flip-flops. Moreover, it performed clearer results in terms of power consumption, delay, and others.

Figure 7 shows the graph of pre-simulation and post-simulation results of the proposed LFSR in terms of total chip power dissipation versus Voltage and current versus Voltage. The pre-simulation total power dissipation is decreased compared to the post-simulation total power dissipation due to the parasitic values and wires. According to the dynamic power dissipation equation ($P_d = C_L V_{dd}^2 f$), when the supply voltage is increased, then the power dissipation also gets increased. According to the Ohms law, the current is increased because the parasitic values are suppressed.

The results of the proposed LFSR circuit simulation are shown in Table 4, and the results are compared to the existing LFSR circuits as in the literature. Although there are different feature sizes and transistors in the existing LFSR circuits, the performance of the proposed circuit has shown a better power consumption result. In [22], the author used a TPG with an LFSR counter for the spacecraft designs. The power consumption of the circuit is 47.7 mW, which the clock's distribution can determine. Hence, in the proposed LFSR, one clock distribution is designed for all the sequential circuits.

When the proposed circuit is compared to the circuit designed in [22], our proposed circuit has a 49.81 % improvement in power dissipation, 52.95 % in the propagation delay, and 6.3 % in the operating frequency. The proposed circuit has fewer transistors than the circuit in [22] since the proposed circuit transistors are arranged with the reduced critical path.

Niclass et al. [23] designed an array of LFSR counters in which each LFSR counter consumed power of 530 mW. Indeed, the reliability of the circuits was evaluated based on the reduced delays in the proposed design. Hence, the area and delay measured are high. Further in this paper, the low power LFSR with effective performance can be achieved compared to [23]. Due to the critical path reduction and lesser transistors, the proposed LFSR circuit achieves a 23.78% improvement in power dissipation, 27.54% in the propagation delay, and 5.26% in the speed of the circuit designed in [23]. In [24], the array of the counters implemented using the 9-bit counters. The circuit designed in [24] achieved a power dissipation of 50 mW and a reduced delay of 52.84 ns with low noise values. Although the proposed LFSR is implemented for 4-bit, it achieved power dissipation at micro-levels. The design in [24] used many stages of the counter for the testing pattern methodology, whereas our proposed circuit used only four stages to achieve the same results. Our proposed LFSR circuit achieved a 51.20% improvement in power dissipation, 52.84% in the propagation delay, and 3.15% in speed than the design in [24].

Pavia et al. [25] designed the LFSR circuit as a TPG for the SRAM memory. The design is based on the differential buffer structural design and executed in a 130 nm CMOS H-spice platform. The propagation delay can be adjusted by using the clock distribution technique. It used a different platform to design a circuit and used for the SRAM memory circuit. The Mentor Graphics IC design tool is used to design the proposed LFSR circuit, and it can achieve 89.62% improvement in the power dissipation, 64.25% in the propagation delay, and 34.21% in the speed as compared to the design in [25]. Daniel et al. [26] used the LFSR counter for large-scale array decoding logic and implemented using the H-spice 130 nm CMOS technology. The authors' design consumed power of 157.5 µW, which is quite high as compared to the proposed LFSR.

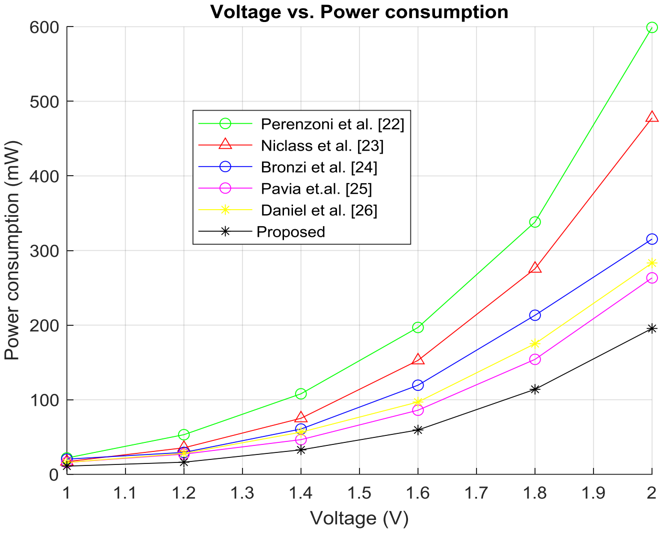


Fig. 8 - Plots of the BIST ref. [28] performance in terms of Power consumption vs. Voltage with the use of proposed LFSR and ref. [22]–[26] LFSR

The proposed work achieved better results for the power and the delay due to the reduced critical path in the transistor and the reduced logical transition in the bit-wise pattern. The proposed algorithm has reduced the critical path between the input and the output, and it gives a new way of design in the LFSR circuit. Moreover, the current density of the N-type transistor is reduced because of the fair sharing of the electron at the output node. Thus, power consumption is reduced in the design of the cell. The power consumption achieved by the proposed LFSR is 25.13 µW and the delay calculated is 49.9 µs. Fig. 8, plotted for the power consumption values for the Elham *et al.* [28] BIST with the proposed LFSR and the existing authors'. The LFSR is used for the test pattern generation in the BIST designs. It is clearly shown that the proposed LFSR consumed lesser power during the BIST operation mode. Whereas, Perenzoni *et al.* [22] LFSR in the BIST ref. [28], consumed higher power of 599.05 µW at the applied Voltage of 2 V compared with the other existing LFSRs'. At the same Voltage of 2V, the proposed LFSR in the BIST achieved the power consumption of 194.52 µW, far better than the Perenzoni *et al.* [22] LFSR.

Conclusion

The paper presented the new LFSR circuit based on the proposed RS algorithm. The practical implementation of the RS algorithm into an LFSR circuit is designed, verified, and analyzed in the BIST design. The overall circuit designs are implemented by CMOS TTL configuration, and the results are analyzed in terms of the area, the power, the delay, and the frequency. The proposed LFSR achieved the maximum length TPGs with re-seeding pattern generation in the BIST design and consumed lower power, lesser delay in the circuits. The power consumption of the pre-simulation as 25.13 μW is measured along with the post-simulation as 34.8 μW at the operating frequency of 1.9 GHz, which could be achieved better performance in the IC designs. Furthermore, the circuit's power consumption could be decreased using the reduced transistors with the low power asynchronous clocks. However, more transistors are included to avoid the swing restoration and signal degradation in the circuits. The paperwork would be designed in applications of BISTs such as spacecraft, musical, and medical instruments for their precise randomness.

Acknowledgment

This work was supported in part by Multimedia University under the Grant number of MMUI/190002.02.

References

[1] D. Xiang, X. Wen, and L. T. Wang, "Low-Power Scan-Based Built-In Self-Test Based on Weighted Pseudorandom Test Pattern Generation and Reseeding," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 3, pp. 942–953, 2017, doi: 10.1109/TVLSI.2016.2606248

[2] B. Mishra, R. Jain, and R. Saraswat, "Low power BIST based multiplier design and simulation using FPGA," *2016 IEEE Students' Conf. Electr. Electron. Comput. Sci. SCEECS 2016*, pp. 2–7, 2016, doi: 10.1109/SCEECS.2016.7509284

[3] H. Mo and M. P. Kennedy, "Masked Dithering of MASH Digital Delta-Sigma Modulators with Constant Inputs Using Multiple Linear Feedback Shift Registers," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 64, no. 6, pp. 1390–1399, 2017, doi: 10.1109/TCSI.2017.2670365

[4] L. J. Weng, "Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms," *IEEE Trans. Comput.*, vol. 43, no. 5, pp. 560–568, 1994, doi: 10.1109/12.280803

[5] P. Alfke, "Efficient Shift Registers, LFSR Counters, and Long Pseudo- Random Sequence Generators," *Xilinx*, vol. 1996, pp. 1–6, 1996, [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp052.pdf

[6] S. Hong and R. Wu, "On deep holes of generalized reed-solomon codes," *AIMS Math.*, vol. 1, no. 2, pp. 96–101, 2016, doi: 10.3934/Math.2016.2.96

[7] M. Chen *et al.*, "Improved BER Performance of Real-Time DDO-OFDM Systems Using Interleaved Reed-Solomon Codes," *IEEE Photonics Technol. Lett.*, vol. 28, no. 9, pp. 1014–1017, 2016, doi: 10.1109/LPT.2016.2523268

[8] C. Yu and Y. S. Su, "Two-mode reed-solomon decoder using a simplified step-by-step algorithm," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 62, no. 11, pp. 1093–1097, 2015, doi: 10.1109/TCSII.2015.2456094

[9] T. Shongwe, "Analysis of the Probability of Sync-Words in Reed-Solomon Codes," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 36–39, 2017, doi: 10.1109/LCOMM.2016.2618370

[10] X. Peng, W. Zhang, W. Ji, Z. Liang, and Y. Liu, "Reduced-Complexity Multiplicity Assignment Algorithm and Architecture for Low-Complexity Chase Decoder of Reed-Solomon Codes," *IEEE Commun. Lett.*, vol. 19, no. 11, pp. 1865–1868, 2015, doi: 10.1109/LCOMM.2015.2477495

[11] H. Dau, I. M. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed-Solomon Codes With Multiple Erasures," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6567–6582, 2018, doi: 10.1109/TIT.2018.2827942

[12] V. T. Van, S. Mita, J. T. Li, C. Yuen, and Y. L. Guan, "Bit-level soft-decision decoding of triple-parity reed-solomon codes through automorphism groups," *IEEE Commun. Lett.*, vol. 17, no. 3, pp. 553–556, 2013, doi: 10.1109/LCOMM.2013.020513.130016

[13] V. Guruswami, S. Member, and M. Wootters, "Repairing Reed-Solomon Codes," vol. 63, no. 9, pp. 5684–5698, 2017

[14] N. Brandonisio *et al.*, "Burst-mode FPGA implementation and error location analysis of forward error correction for passive optical networks," *J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 298–308, 2018, doi: 10.1364/JOCN.10.000298

[15] S. J. Lin, T. Y. Al-Naffouri, and Y. S. Han, "FFT Algorithm for Binary Extension Finite Fields and Its Application to Reed-Solomon Codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5343–5358, 2016, doi: 10.1109/TIT.2016.2600417

[16] S. Kopparty, N. Ron-Zewi, S. Saraf, and M. Wootters, "Improved decoding of folded reed-solomon and

multiplicity codes," *Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS*, vol. 2018-Octob, pp. 212–223, 2018, doi: 10.1109/FOCS.2018.00029

[17] A. Shokrollahi, "A class of generalized RS-codes with faster encoding and decoding algorithms," *2013 Inf. Theory Appl. Work. ITA 2013 - Conf. Proc.*, pp. 82–91, 2013, doi: 10.1109/ITA.2013.6502932

[18] marvin, “Coding Concepts and Reed-Solomon Codes,” pp. 1–222, 2014, [Online]. Available: papers3://publication/uuid/F7846238-D56B-4736-A695-9C4F56006763

[19] O. Acevedo and D. Kagaris, "On the computation of LFSR characteristic polynomials for built-in deterministic test pattern generation," *IEEE Trans. Comput.*, vol. 65, no. 2, pp. 664–669, 2016, doi: 10.1109/TC.2015.2428697.

[20] R. R. C. Over, "Gf (2)," vol. 24, no. 1, pp. 2019–2022, 2020

[21] J. M. Zhang, W. F. Qi, T. Tian, and Z. X. Wang, "Further Results on the Decomposition of an NFSR into the Cascade Connection of an NFSR into an LFSR," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 645–654, 2015, doi: 10.1109/TIT.2014.2371542

[22] M. Perenzoni, D. Perenzoni, and D. Stoppa, "A 64×64-pixel digital silicon photomultiplier direct ToF sensor with 100Mphotons/s/pixel background rejection and imaging/altimeter mode with 0.14% precision up to 6km for spacecraft navigation and landing," *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 59, pp. 118–119, 2016, doi: 10.1109/ISSCC.2016.7417935

[23] C. Niclass, M. Soga, H. Matsubara, M. Ogawa, and M. Kagami, "Depth Sensor," vol. 49, no. 1, pp. 1–16, 2014.

[24] D. Bronzi *et al.*, "Array for 2-D Imaging and 3-D Ranging," *IEEE J. Sel. Top. Quantum Electron.*, vol. 20, no. 6, 2014

[25] J. M. Pavia, M. Scandini, S. Lindner, M. Wolf, and E. Charbon, "A 1 × 400 Backside-Illuminated SPAD Sensor with 49.7 ps Resolution, 30 pJ/Sample TDCs Fabricated in 3D CMOS Technology for Near-Infrared Optical Tomography," *IEEE J. Solid-State Circuits*, vol. 50, no. 10, pp. 2406–2418, 2015, doi: 10.1109/JSSC.2015.2467170

[26] D. Morrison, D. Delic, M. R. Yuce, and J. M. Redoute, "Multistage Linear Feedback Shift Register Counters with Reduced Decoding Logic in 130-nm CMOS for Large-Scale Array Applications," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 27, no. 1, pp. 103–115, 2019, doi: 10.1109/TVLSI.2018.2872021

[27] A. Brown, "Implementing Reed-Solomon," *Duke Univ.*, pp. 1–37, 2011, [Online]. Available: https://www2.cs.duke.edu/courses/spring11/cps296.3/decoding_rs.pdf

[28] E. Moghaddam, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer, and J. Zawada, "Logic BIST with Capture-Per-Clock Hybrid Test Points," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 38, no. 6, pp. 1028–1041, 2019, doi: 10.1109/TCAD.2018.2834441