Investigation of Some Quite Interesting Divisibility Situations in a Signature Analyzer Implementation

AFAQ AHMAD
Department of Electrical and Computer Engineering
College of Engineering
Sultan Qaboos University
P.O. Box – 33, Zip Code – 123
OMAN
afaq@squ.edu.om; afaqahmad51@gmail.com
http:// www.squ.edu.om

Abstract: - When designing error detecting and correcting systems, cryptographic apparatus, scramblers and other secure, safe and authenticated communication and digital system response data compression devices, the division of polynomials are frequently involved. Commonly, the process of division is implemented by using hardware known as Linear Feedback Shift Registers (LFSRs). In digital system testing the technique of Built-In Self Test (BIST) uses this LFSR based division process for response data compression and is popularly known as Signature Analyzer (SA). This paper presents a simulation experiment on the effectiveness study of the SA schemes. The finding of the results of the simulation study reveals that in SA implementation; in general the uses of primitive characteristic polynomials are the best. However, the study further investigates that the use of some critical primitive characteristic polynomials may reverse the effectiveness of the SA schemes i.e. lead to observe maximum aliasing errors.

Key-Words: - Signature Analyzer, Linear Feedback Shift Registers, Built-In Self-Test, VLSI, Aliasing Errors, Characteristic Polynomial, Primitive Polynomials, Polynomial Division, Cyclic Redundancy Check

1 Introduction

Because of its many inherent advantages, currently, Built-In Self-Test (BIST) has become an effective and widely acceptable tool for tackling test problems for VLSI chips and digital systems [1-6]. By building test circuitry on chip, BIST techniques usually combine a built-in stimulus source (test sequence generator) with a response data compressor. This approach eliminates the complex task of integrating separate circuits for test-pattern generation and response data compression. Besides, BIST approach minimizes the storage requirements of test sequences and large response data, as well as reduces the test time, and isolates defect to chip level itself. Furthermore, because test stimuli are applied using normal clock rate, testing of a self-testable chip can be performed at-speed. Additionally, since the test resources are available during the entire life of the chip, they can significantly simplify the diagnostics and maintenance procedures for digital systems [1 - 8].

In built-in self-test environment Linear Feedback Shift Registers (LFSRs) is an integral part of sequential design, such that they can be used for both generating the test sequences and compressing the output response data using SA scheme (see Fig. 1).

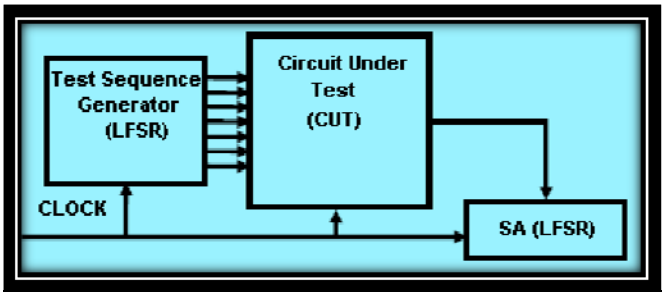

Figure 1: BIST approach of testing

But the difficulty arises when the resulting response data obtained from the Circuit Under Test (CUT) is compressed into small signatures using Signature Analyzer (SA) via response data compression tool. Although, SA scheme is easily implemented by an LFSR, but this leads to loss of information, due to the erroneous response patterns that gets compressed into the same signature as the fault free signature of the CUT. Thus, some of the faults might go undetected due to this error-masking

phenomenon. Therefore, this compression technique can further reduce the fault-coverage in the BIST scheme. This particular problem of error masking is called 'aliasing' phenomenon in the field of digital system testing [1 - 14].

Methods to determine the extent of error-masks caused by a response data compressor are not readily available. However, various attempts [1 – 25] been made to analyze and improve the basic signature scheme. The end goal of these schemes, individually, or with a combination of these, is to reduce the deception volume. In the research papers it is conjectured that the changed order of the test patterns applied to the CUT may change the level of the probability of error masking behaviour. Whereas some of the simulation results via one of the communications have been demonstrated that the changes in the polynomial seed ('initial loading of LFSR') in LFSR based BIST technique do not affect the error-masking behaviour of an SA scheme. This paper further investigates the SA schemes and presents the results through a developed simulation model which demonstrates that there exists a special relationship between the pairs of primitive characteristic polynomials of the SA. It is found that in a circumstance the use of a primitive characteristic polynomial may be the best effective but at the same time the use of the reciprocal of the same primitive characteristic polynomial may prove the effectiveness of SA as the worst one.

2 LFSR Theory

The theory of LFSR and related issues are readily available in the literatures [26 – 34]. Just to make this paper more readable we reintroduce the available theory of LFSR in brief. Fig. 2 depicts a general model of an n-bit LFSR realized by an external exclusive OR bank. An LFSR has two components: a shift register (or the law describing shifts in each bit which uses D flip-flops) and a feedback function (the new bit can be translated to an actual number or a useful message; realized by a bank of exclusive OR function). It is simply a sequence of different bits created by shifting 1-bit to the right. The extreme left bit is obtained as a function of the other bits in the register depending on the feedback taps [c0, c1, c2, . . . , cn-1, cn]. In an LFSR connection, the number of cells tapped determines the polynomial that characterizes the entire connection of the LFSR. Obviously, the last cell is always tapped (cn = 1). The characteristic equation given in equation (1) always starts with x^n ($1*x^n = x^n$) and has a one at the end since the feedback connection starts with the cell 0 and therefore, c0 will always 1 and $x^o = 1$, hence 1*1 gives 1. The cells that are tapped are linked to XOR operation giving a feedback of different bits every time. Consider the following example elaborated through Table 1. Table 1 illustrates that how the patterns are produced by the LFSR having feedback taps from 1st and 3rd (c0 = c1 = c3 = 1; characteristic polynomial $P(x) = 1 + x + x^3$) also, assume that the pattern of 111 is used as an initial loading (seed) of LFSR. It can be visualized as the pattern 8 becomes as the pattern 1, repetition starts thus the period of LFSR sequence is 7. Equation (1) denotes the governing equation of LFSR characteristic polynomial P(x).

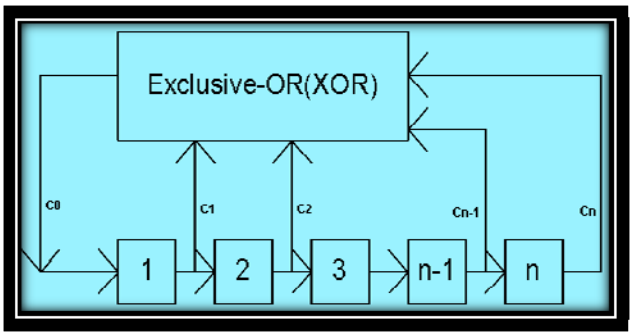

Figure 2: An n-bit LFSR

Table1: LFSR pattern for $P(x) = 1 + x + x^4$

Clock	Bit-1	Bit-2	Bit-3	Bit-4
1	1	1	1	1
2	0	1	1	1
3	1	0	1	1
4	0	1	0	1
5	1	0	1	0
6	1	1	0	1
7	0	1	1	0
8	0	0	1	1
9	1	0	0	1
10	0	1	0	0
11	0	0	1	0
12	0	0	0	1
13	1	0	0	0
14	1	1	0	0
15	1	1	1	0
16	1	1	1	1

$$P(x)=1+\left(\sum_{i=1}^{n-1} c_i x^i\right)+x^n \quad (1)$$

As shown in Equation 2, that the transition matrix (represented by [A] of order nxn of an n-

stage LFSR) which, if multiplied by the initial or the present state Y(t) of the LFSR gives the state of the LFSR one clock later or the next state Y(t+1). The matrix A is demonstrated in Equation (3).

$$[Y(t+1)] = [A]\,[Y(t)] \quad (2)$$

$$\begin{bmatrix} y_1(t+1) \\ y_2(t+1) \\ \vdots \\ y_{n-1}(t+1) \\ y_n(t+1) \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_{n-1}(t) \\ y_n(t) \end{bmatrix} \quad (3)$$

Where $c_j = 0$ or 1, for $1 \le j \le n-1$ shows absence or presence of the respective feedback taps; $c_j = 1$, for j = n to complete the feedback connection for the entire LFSR.

The period p of the shift register is the number of new bits created without repetition i.e. [Y (t+p)] = [Y (t+1)]. A sequence generated from an n-stage LFSR has a maximal length of 2^n-1. However, if we start with a shift register filled with all zeros the LFSR will generate a never-ending stream of zeros i.e. the period of LFSR sequence becomes zero.

Not all (2^{n-1}) the characteristic polynomials (corresponding to feedback taps of an n-bit LFSR) would be capable of generating maximal length sequence. There are many properties of polynomials which generate maximal length sequence. Like only the n^{th} order primitive characteristic polynomials can generate the sequence of length 2^n-1. Some of such properties related to LFSR are summarized below in the form theorems and definitions.

Theorem 1:

In an n-bit LFSR, the total number of possible characteristic polynomials (NPP) will be equal to 2^{n-1}.

Proof:

Since the last bit of the LFSR is always tapped, and the taps (c_i has binary option either 0 or 1, representing the absence and presence of the taps respectively) therefore, NPP = 2^{n-1}.

Definition 1:

The period p of an n-bit LFSR is the length of the cycle after which the LFSR sequence repeats.

Definition 2:

The period p of an n-bit LFSR will only be maximal when p = m = 2^n-1.

Definition 3:

A sequence produced by an n-bit LFSR which has period m is called a PN-sequence (or a pseudo-noise sequence or m-sequence or maximal length sequence).

Definition 4:

In an n-bit LFSR model a characteristic polynomial is characterized as an irreducible (which cannot be factored) has a period p which divides m (p < m).

Definition 5:

In an n-bit LFSR model an irreducible characteristic polynomial is characterized as a primitive polynomial whose period is m (p = m).

Theorem 2:

In an n-bit LFSR, a characteristic polynomial P(x) is primitive, if and only if, it's reciprocal characteristic polynomial $P^*(x)$ is also primitive.

The reciprocal characteristic polynomial of P(x), denoted by P*(x) and can be given as shown in Equation (4).

$$P^*(x) = x^n P(1/x) = c_0 x^n + c_1 x^{n-1} + \ldots + c_n \quad (4)$$

That is if P(x) corresponds to the taps {n, .., i, …, j, …} then, P*(x) will correspond to the taps {n, … n-i, …, n-j, …} where n > i > j.

Theorem 3 below which is consequence of Theorem 2 can be interpreted as:

Theorem 3:

The characteristic polynomial of an n-bit LFSR can be primitive, if and only if the number of taps in that LFSR is even excluding the tap c_0.

The determination of the primitive polynomial involves the use of the Euler phi-function Φ(.) and search for primes. The Euler function has the property that its value for an integer m is the product of the values of the Euler phi-function at the prime powers that occur in the factorization of m. The Euler phi-function is computed on the basis of the prime factorization of m (m = p). The following equation (Equations (5) and (6)) are embodied in the proposed algorithm for finding primitive polynomials.

The number of possible primitive polynomials (NPP) of order n can be found out by using Euler's phi-function Φ(m) as stated below.

Let $m = p_1^{a_1} p_2^{a_2} \cdots\cdots p_k^{a_k}$ be the prime-power factorization of a positive integer m. Then

$$\Phi(m) = m\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_k}\right) \quad (5)$$

The NP can be computed as in Equation 5.

$$NPP = \frac{\Phi(m)}{n} \quad (6)$$

Let us consider n = 3, there exists $2^{3-1} = 4$, total possible polynomials, $m = 2^n - 1 = 7$, so, m is prime therefore, $\phi(m) = 6$, whereas, NPP will be equal to 2. Let us consider another example of n = 4 which has 8 possible characteristic polynomials, m = 15 ($p_1 = 3$, $p_2 = 5$), $\phi(m) = 8$ gives NPP = 2.

Golomb's [33] principles related to maximal length sequence are nothing but known as the properties of the sequence and, described below:

- An n -bit LFSR generates a maximal length sequence of period 2^n-1.
- A maximal length sequence produces exactly $2^{(n-1)}$ total ones and $2^{(n-1)}$-1 total zeros in the sequence.
- The run length of a maximal length sequence will have

→ 1 run of ones of length n, and 1 run of zeros of length n-1.

→ 1 run of ones and 1 run of zeros, each of length n-2.

→ 2 runs of ones and 2 runs of zeros, each of length n-3

→ 4 runs of ones and 4 runs of zeros, each of length n-4.

Table 2 provides the data to demonstrate the properties of maximal length sequences of LFSRs of sizes 2 to 5.

Table 2: Different runs of maximal length sequences

LFSR size / P(x) / p	Sequence consists of									
	5-1's	5-0's	4-1's	4-0's	3-1's	3-0's	2-1's	2-0's	1-1's	1-0's
2/ 1+x +x² / 3 Total number of 1's = 2, 0's = 1	X	X	X	X	X	X	1	0	0	1
3/ 1+x² +x³ / 7 Total number of 1's = 4, 0's = 3	X	X	X	X	1	0	0	1	1	1
4/ 1+x³ +x⁴ / 15 Total number of 1's = 8, 0's = 7	X	X	1	0	0	1	1	1	2	2
5/ 1+x² +x⁵ / 31 Total number of 1's = 16, 0's = 15	1	0	0	1	1	1	2	2	4	4

3 Signature Analysis Process

Signature Analyzer is a response data compression tool based on the concept of Cyclic Redundancy Checking (CRC). The LFSRs are used to realize a SA in hardware form. The first signature analyzer, HP5004A, was manufactured by Hewlett-Packard [35]. The SA is used to detect errors in data streams caused by the faults in a CUT. The simplest form of the SA consists of a single-input LFSR and is known as Single Input Shift Register (SISR). A SISR type SA with internal exclusive OR arrangement is shown in Fig. 3. In the operation of such SA schemes the signature is then the contents of the register after the last data input bit has been sampled. The Table 3 demonstrates the process of obtaining the signature vector [s] for a given data vector [d] = [1001011], using a 3-bit LFSR circuit with feedback connections (CON) as CON = [c_0, c_1, c_2, c_3] = [1, 1, 0, 1].

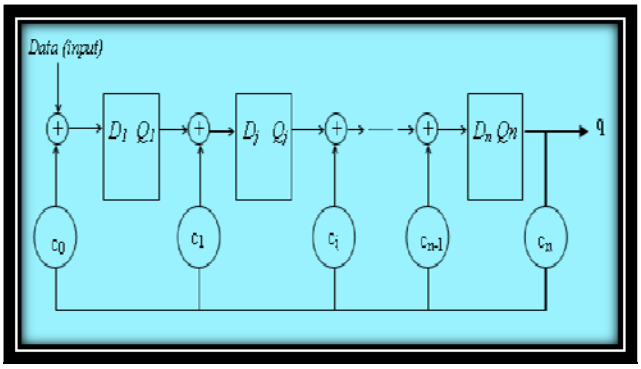

Figure 3: An n-bit SISR type SA

Table 3: The state table of a 3-bit SA

Clock	Register's states			Comments
	Q1	Q2	Q3	
t = 0	0	0	0	Initialization of SA
t = 1	1	0	0	The MSB entered '1'
t = 2	0	1	0	The bit entered '0'
t = 3	0	0	1	The bit entered '0'
t = 4	0	1	0	The bit entered '1'
t = 5	0	0	1	The bit entered '0'
t = 6	0	1	0	The bit entered '1'
t = 7	1	0	1	The LSB entered '1'
t = 7	1	0	1	The signature [s]

The SA process can be defined as a polynomial division in GF(2) of the data polynomial d(x) by the polynomial P(x), which is a function of the feedback coefficients of LFSR circuit of the SA. The polynomial P(x) is conventionally referred to as the characteristic polynomial of the SA. The signature polynomial s(x) is simply the remainder of the polynomial division of d(x) over P(x). The

Equation (7) describes the division process. In Equation (7), the q(x) is the quotient polynomial. The Example 1 below demonstrates the polynomial division process of SA.

$$\frac{d(x)}{P(x)} = q(x) + \frac{s(x)}{P(x)} \tag{7}$$

The SA process can also be defined by using state space model as given in Equation (8).

$$[Y(t+1)] = [A]*[y(t)] + [d] \tag{8}$$

To make it more readable we present the example as below.

Example 1:

Let us consider the same data as considered for the Table 3. i.e. [d] = [1001011], using a 3-bit LFSR circuit with feedback connections (CON) as CON = [c_0, c_1, c_2, c_3] = [1, 1, 0, 1].

Then d(x) = 1 + x + x^3 + x^6 and P(x) will be as P(x) = 1 + x + x^3.

Using Equation (7), we get
s(x) = 1 + x^2, and
q(x) = x + x^3

To verify the results which are obtained by the LFSR circuit and the polynomial division process the Test1, Test2, and Test3 are the procedures of polynomial multiplication, polynomial division, and binary division processes in GF(2), are described below respectively.

Test 1:

Using Polynomial Multiplication Process in field GF(2)

$$\begin{aligned} q(x)\,P(x) + s(x) &= d(x) \\ &= (x + x^3)(1 + x + x^3) + 1 + x^2 \\ &= 1 + x + 2x^2 + x^3 + 2x^4 + x^6 \\ &= 1 + x + x^3 + x^6 = d(x) \end{aligned}$$

This verifies the model of SA process.

Test 2:

Figure 4 demonstrates the verification of Example 1 using Polynomial Division Process in GF(2).

Similarly, Test 3 below verifies and demonstrates Binary Division

Test 3:

Figure 5 demonstrates the verification of Example 1 using Binary Division Process

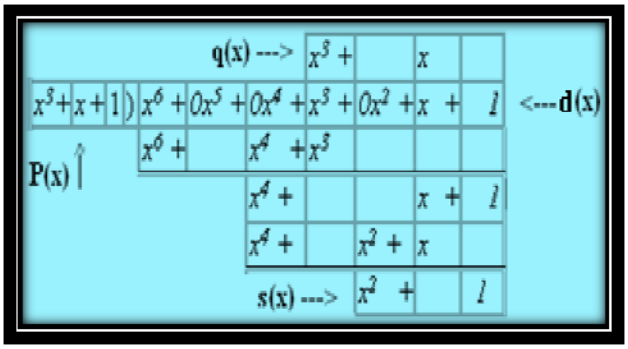

Figure 4: Test 2; Polynomial Division Process

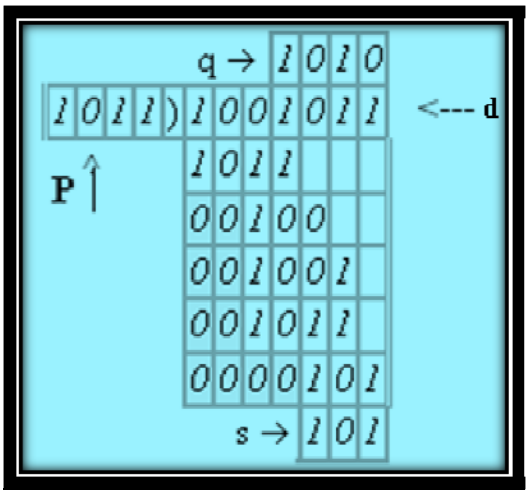

Figure 5: Test 3; Binary Division Process

The above Tests demonstrates the results are having the conformity with the results of the LFSR based SA circuit. It can be seen in the Table 3 that at t = 7, [1 0 1] is the signature vector s whereas, the vector [0 0 1 0 10], obtained from the output Q_3, of the register D_3 collected before the last bit of the data is entered (i.e. up to t = 6) is the quotient q.

4 Simulation Set-Ups

Data (input) to the SA in Fig. 6 is maximal length sequence (m-sequence) generated by the respective possible primitive polynomials, MGP(x) of order n. The m-sequence generator model used in this set-up is shown in Fig. 4. This m(x) is applied to CUT (see Fig. 1). The divisibility of SA is tested for each of the m-sequence vectors through all possible primitive characteristic polynomials of order n. The procedure adapted in this study is summarized below in the form of an algorithm.

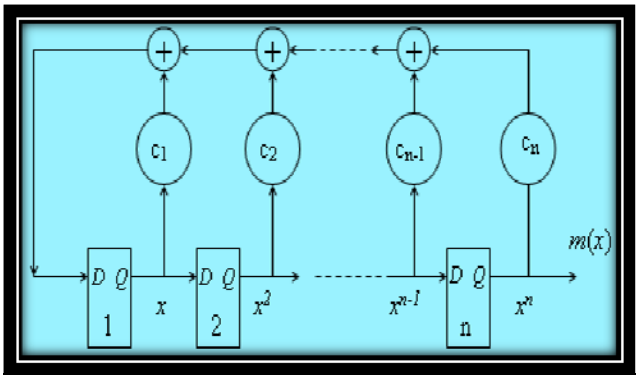

Figure 4: An n-bit LFSR – sequence generator

ALGORITHM

Begin
STEP 0:
For n = 3: N
% (for this communication the N is used as 6)
STEP 1:
Generate and list all possible primitive polynomials of order n, ($P_{i,n}$ (x))
% (where i vary from 1 to NPP; the NPP is the total %number of primitive polynomials of order n)
STEP 2:
Obtain and list the pairs of reciprocal polynomials from the list of NPP number of primitive polynomials of order n
STEP 3:
For j = 1: NPP
Generate and document the binary and hex forms of m-sequence by using the generator polynomial $P_{j,n}(x)$
For k = 1: NPP
Check the divisibility of SA by using the characteristic polynomial $P_{k,n}(x)$ while inputting the m-sequence generated by polynomial $P_{j,n}(x)$;
End;
Document the divisibility result (the quotient polynomials q(x) and the remainder polynomial s(x));
End;
End

5 Simulation Run

Hence using the algorithm described in Section 4, the obtained data are presented through Tables 4 – 8. The Table 4 lists the primitive polynomials and their respective reciprocal pairs. For acquiring more knowledge about findings of such pairs one can refer to research papers [31], [32], and [36]. Whereas, Tables 5 - 8 are demonstrating the SA divisibility with respect to all possible combinations of $P_{j,n}(x)$ and $P_{k,n}(x)$. The quotients are presented in the tables only for n = 3 and 4. Although the simulation experiment provides all the q(x) values however, it is difficult to present for the large sizes of n.

Table 4: Primitive polynomials and reciprocal pairs

n / NPP	Primitive polynomials P(x)	reciprocals
3 / 2	$P_{13} = 1 + x^2 + x^3$	P_{13}, P_{23}
	$P_{23} = 1 + x + x^3$	
4 / 2	$P_{14} = 1 + x^3 + x^4$	P_{14}, P_{24}
	$P_{24} = 1 + x + x^4$	
5 / 6	$P_{51} = 1+x^3+x^5$	P_{51}, P_{52}
	$P_{52} = 1+x^2+x^5$	P_{53}, P_{56}
	$P_{53} = 1+x^2+x^3+x^4+x^5$	P_{54}, P_{55}
	$P_{54} = 1+x+x^3+x^4+x^5$	
	$P_{55} = 1+x+x^2+x^4+x^5$	
	$P_{56} = 1+x+x^2+x^3+x^5$	
6 / 6	$P_{61} = 1+x^5+x^6$	P_{61}, P_{63}
	$P_{62} = 1+x^2+x^3+x^5+x^6$	P_{62}, P_{65}
	$P_{63} = 1+x+x^6$	P_{64}, P_{66}
	$P_{64} = 1+x+x^4+x^5+x^6$	
	$P_{65} = 1+x+x^3+x^4+x^6$	
	$P_{66} = 1+x+x^2+x^5+x^6$	

Table 5: Divisibility measure for SA (n = 3)

m-sequence generator polynomial $P_{j,n}(x)$	m-sequence vector (m)	Divisibility of m with different SA polynomials $P_{k,n}(x)$ Remainders: polynomial		Quotient: polynomial with different	
		P_{13}	P_{23}	P_{13}	P_{23}
P_{13}	[1001011]	0	$1 + x^2$	q_{13}	q_{23}
P_{23}	[1001110]	$1 + x^2$	0	q_{33}	q_{43}

$q_{13} = +x+x^2+x^3$; $q_{23} = x+x^3$;
$q_{33} = x+x^3$; $q_{43} = 1+x+x^2+x^3$

Table 6: Divisibility measure for SA (n = 4)

m-sequence generator polynomial $P_{j,n}(x)$	m-sequence in binary (m)	Divisibility of m with different SA polynomial $P_{k,n}(x)$ Remainder polynomial		Quotient: polynomial with different SA polynomials	
		P_{14}	P_{24}	P_{14}	P_{24}
P_{14}	m_1	0	$1 + x^2 + x^3$	q_{14}	q_{24}
P_{24}	m_2	$1 + x^2 + x^3$	0	q_{34}	q_{44}

m_1 = [100010011010111];
m_2 = [100011110101100]
$q_{14} = 1+x+x^2+x^3+x^4+x^6+x^7+x^8+x^9+x^{10}$;
$q_{24} = x+x^2+x^4+x^7+x^{10}$;
$q_{34} = 1+x^3+x^4++x^5+x^6+x^7+x^8+x^9+x^{10}$;
$q_{44} = x^2+x^5+x^7+x^{10}$

Table 7: Divisibility measure for SA (n = 5)

m-sequence generator polynomial $P_{j,n}(x)$	m-sequence (m) in Hexadecimal	Divisibility of m with different SA polynomials $P_{k,n}(x)$ Remainders: in Hexadecimal					
		P_{51}	**P_{52}**	**P_{53}**	**P_{54}**	**P_{55}**	**P_{56}**
P_{51}	**4259F1BA**	**0**	**19**	**0**	**0**	**0**	**0**
P_{52}	**42BB1F34**	**14**	**0**	**0**	**0**	**0**	**0**
P_{53}	**42D477C9**	**0**	**0**	**0**	**0**	**0**	**1D**
P_{54}	**439BE895**	**0**	**0**	**0**	**0**	**15**	**0**
P_{55}	**43522FB3**	**0**	**0**	**0**	**1D**	**0**	**0**
P_{56}	**4327DC56**	**0**	**0**	**11**	**0**	**0**	**0**

Table 8: Divisibility measure for SA (n = 6)

m-sequence generator polynomial $P_{j,n}(x)$	m-sequence (m) in Hexadecimal	Divisibility of m with different SA polynomials $P_{k,n}(x)$ Remainders: in Hexadecimal					
		P_{61}	P_{62}	P_{63}	P_{64}	P_{65}	P_{66}
P_{61}	m_1	0	0	31	0	0	0
P_{62}	m_2	0	0	0	0	29	0
P_{63}	m_3	31	0	0	0	0	0

m_1 = 410C53D1C96ECD5F;
m_2 = 417E5467BAD36223;
m_3 = 41FAB376938BCA30;
m_4 = 41E4A9A116FD719D;
m_5 = 41C246CB5DE62A7E;
m_6 = 41B98EBF68859527

6 Results and Analysis

Analyzing the results through Tables 5-8 given in Section 5, the following findings are summarized and presented in the forms of theorems.

Theorem 4:

The m-sequence generated by an n-bit primitive characteristic polynomial P(x) will be divisible if the same characteristic polynomial P(x) is used in the signature analysis scheme.

In Tables 5 - 8 it can be seen that when the m-sequence generator polynomial and the SA polynomial are same the values of the remainders are zeros i.e. the input sequence polynomials are exactly divisible by its respective SA polynomials.

Theorem 5:

The m-sequence generated by an n-bit primitive characteristic polynomial P(x) will not be divisible if the reciprocal of characteristic polynomial P(x) is used in the signature analysis scheme.

It can be seen in Tables 3 - 6 that when the m-sequence generator polynomial and the SA polynomial are reciprocals the values of the remainders are not zeros i.e. the input sequence

polynomials are not divisible by its respective reciprocal polynomials used in SA.

7 Conclusion

The effectiveness of any SA scheme depends upon following factors,

i. The size of SA scheme n,
ii. The characteristic polynomial of SA tool, and
iii. The nature of the inputted data stream into the SA scheme.

The findings through this paper clearly, indicates that the use of primitive characteristic polynomials in SA scheme is intended to give the best results however, it can also be critically analyzed that all the primitive characteristic polynomials will not provide the same effectiveness for the SA scheme. As it can be seen from the Tables 5 - 8 that the use of reciprocals of primitive polynomials results the totally different situations of divisibility of the SA scheme which turns from divisible to not divisible. In Table 7 we observe that the m-sequence (410C53D1C96ECD5F) given in hexadecimal base is generated by a characteristic primitive polynomial, P_{61}; which is represented by the polynomial $1+x^5+x^6$ is divisible by all other possible primitive polynomials except P_{63} which is reciprocal of P_{61}. When P_{63} is used as characteristic polynomial in the SA then the data denoted in hexadecimal as (410C53D1C96ECD5F) gives the signature as (31) given in hexadecimal notation hence not divisible. We obtained the results for n = 3 to 16 and similar findings exist. This finding provides a breakthrough in the research of the study of the effectiveness of SA schemes.

ACKNOWLEDGEMENTS

The acknowledgements are due to authorities of Sultan Qaboos University (Sultanate of Oman) for providing generous research support grants and environments for carrying out the research works.

References:

[1] Ahmad, A., "A Simulation Experiment on a Built-In Self Test Equipped with Pseudorandom Test Pattern Generator and Multi-Input Shift Register (MISR)", *International Journal of VLSI Design & Communication Systems*, vol. 1, no. 4, pp. 1-12, 2010

[2] Ahmad, A., Dawood Al-Abri, "Design of an Optimal Test Simulator for Built-In Self Test Environment", *Journal of Engineering Research*, vol. 7, no. 2, pp. 69 – 79, 2010

[3] Ahmad A., 'Testing of Complex Integrated Circuits (ICs) – The Bottlenecks and Solutions', *Asian Journal of Information Technology*, vol. 4, no. 9, pp. 816 – 822, 2005

[4] Ahmad, A. and Al-Habsi, A. H., "Design of a Built-In Multi-Mode ICs Tester with Higher Testability Features- A Most Suitable Testing Tool for BIST Environment," *Journal of IETE Technical Review*, vol. 15, no. 3, pp. 283 – 288, 1998

[5] Nanda N.K., Ahmad A. and Gaindhar V.C., "Shift Register Modification for Multipurpose Use in Combinational Circuit Testing," *International Journal of Electronics (UK)*, vol.66, no.6, pp. 875 – 878, 1989

[6] E. J. McCluskey, 'Built-In Self-test Techniques', *IEEE Design & Test of Computers*, vol. 2, no. 2, pp. 21-28,1985

[7] Ahmad A. and Nanda N.K., "Effectiveness of Multiple Compressions of Multiple Signatures," *International Journal of Electronics (UK)*, vol.66, no.5, pp.775 – 787, 1989

[8] Ahmad A., "Achievement of Higher Testability Goals Through the Modification of Shift Register in LFSR Based Testing," *International Journal of Electronics (UK)*, vol. 82, no. 3, pp. 249-260, 1997

[9] Ahmad A., Al-Lawati A. M. J., Jervase J. A. and Zabalawi, I. H., "The Study of the Effect of Rotationally Delayed Transmission of Data on Error Masking Behaviour of Different Types of Signature Analysis Schemes," *Journal for Scientific Research - Science and Technology*, vol. 1, p. 88, 1996

[10] V. N. Yarmolic, 'On the Validity of Binary Data Sequence by Signature Analyzer', *Electron Model*, vol. 6, pp. 49-57, 1985.

[11] S. G. Akl, 'Digital Signatures - A Tutorial Survey', *IEEE computer*, vol. 16, no. 2, pp. 15-24, 1983

[12] D. W. Davies, 'Applying the RSA Digital Signature to Electronic Mail', *Computer*, vol. 16 no. 2 pp. 55-65, 1983

[13] S. M. Matyas and C. H. Meyer, 'Electronic Signature for Data Encryption Standard', *IBM Technical Bulletin*, vol. 24 no. 5, pp. 2332-34, 1981

[14] K. G. Kiryonov, 'On Theory of Signature Analysis - Communications Equipment', *Radioizm Tekh*, vol. 27, no. 2, pp. 1-46 1980

[15] Ali Al-Lawati and Ahmad, A., "Realization of a Simplified Controllability Computation Procedure – A MATLAB-SIMULINK Based Tool," *Journal for Scientific Research - Science and Technology*, vol. 8, 2004, pp. 131 – 143, 2004

[16] Ahmad A, Al-Lawati, A. M. J. and Ahmed M. Al-Naamany, "Identification of Test Point insertion Location Via Comprehensive Knowledge of Digital System's Nodal Controllability Through a Simulated Tool," *Asian Journal of Information Technology (AJIT)*, vol. 3, no. 3, pp. 142 – 147, 2004

[17] Ahmad, A., 'Investigation of a Constant Behavior Aliasing Errors in Signature Analysis Due to the Use of Different Ordered Test-Patterns in BIST Testing Techniques', *Journal of Microelectronics and Reliability, (PERGAMON, Elsevier Science)*, vol. 42, pp. 967 – 974, 2002

[18] Ahmad, A., 'Constant Error Masking Behavior of an Internal XOR Type Signature Analyzer Due to the Changed Polynomial Seeds," *Journal Of Computers & Electrical Engineering (PERGAMON, Elsevier Science)*, vol. 28, no. 6, pp. 577 – 585, 2002

[19] Al-Naamany, A. M., and Ahmad A., 'Development of a Strong Stream Ciphering Technique Using Non-Linear Fuzzy Logic Selector', *Mobile and Wireless Communications, Kluwer Academic Publishers (Reference: PWC'02 – IFIP 106; Singapore)*, pp. 199 – 206, 2002

[20] Ahmad A., Al-Musharafi. M. J., Al-Busaidi, S., 'Design and Study of a Strong Stream Crypto-System Model for e-Commerce', *International Council for Computer Communication Publishers – Washington DC, USA (The ACM Library)*, pp. 619 – 630, 2002

[21] P. Sarkar, B. K. Roy, P. P. Choudhary and R. Barua, 'Polynomial Division Using Left Shift Register, *Journal of computers & mathematics with applications'*, vol. 35, no. 6, pp. 27-31, 1998

[22] Ahmad A., 'Critical Role of Polynomial Seeds on the Effectiveness of an LFSR-Based Testing Technique', *International Journal of Electronics (UK)*, vol.77, no.2, pp.127 – 137, 1994

[23] Ahmad A., Nanda N.K. and Garg K., 'Are Primitive Polynomials Always Best in Signature Analysis?', *IEEE Design & Test of Computers (USA)*, vol.7, no.4, pp. 36 – 38, 1990

[24] Ahmad A., Nanda N.K. and Garg K., 'A Critical Role of Primitive Polynomials in an LFSR Based Testing Technique', *IEE Electronics Letters (UK)*, vol.24, no.15, pp. 953 – 955, 1988

[25] Ahmad A., Nanda N.K. and Garg K., 'The Use of Irreducible Characteristic Polynomials in an LFSR Based Testing of Digital Circuits', *Proceedings of 4th IEEE int'l conference (TENCON-89)*, held at Bombay (India), Nov. 21-23, pp. 494-496, 1989

[26] Ahmad, A., "Investigation of Typical Properties of Some LFSR Structures," *Journal of System Science and Engineering*, vol. 17, no. 1, pp. 65 – 69, 2008

[27] Ahmad, A., and Al-Maashri, A., 'Investigating Some Special Sequence Length Generated Through an External Exclusive-NOR Type LFSRs,' *International Journal Electrical and Computer Engineering, (PERGAMON, Elsevier Science)*, vol. 34, pp. 270 – 280, 2008

[28] Ahmad, A., "Development of State Model Theory for External Exclusive NOR Type LFSR Structures," *Enformatika*, Volume 10, pp. 125 – 129, 2005

[29] Ahmad, A., Nadir, Z. and Khan, F. A., 'FPGA Based Design of Faster PN Generators for the Use of CDMA Applications', *Proceedings Wireless and Optical Communications Networks (WOCN 2004), IFIP TC6/ IEEE, Sultan Qaboos University Publication*, pp. 272 – 275, 2004.

[30] T. Jamil and Ahmad A., 'An Investigation in to the Application of Linear Feedback Shift Registers for Steganography', *Proceedings IEE SoutheastCon2002*, pp. 239 – 244, 2002.

[31] Ahmad, A., Al-Musharafi, M.J., and Al-Busaidi S., 'A New algorithmic procedure to Test m-Sequences Generating Feedback Connections of Stream Cipher's LFSRs', - *IEEE 01CH37239 (TENCON'01)*, pp. 366 – 369, 201

[32] Ahmad A. and Elabdalla A. M., 'An Efficient Method to Determine Linear Feedback Connections in Shift Registers that Generate Maximal Length Pseudo-Random Up and Down Binary Sequences', *Computer & Electrical Engineering -An Int'l Journal (USA)*, vol. 23, no. 1, pp. 33-39, 1997.

[33] S. W. Golomb, Shift Register Sequences, *Aegean Park Press*, Leguna Hills, U.S.A. 1982

[34] W. W. Peterson and J. J. Weldon, Error correcting codes, 2nd edition, *MIT Press*, Cambridge, London 1972

[35] R. A. Frohwerk, 'Signature Analysis, a New Digital Field Service Method', *Hewlett-Packard Journal*, pp. 2-8, 1977

[36] Ahmad A., and Hayat, L., 'Selection of Polynomials for Cyclic Redundancy Check for the Use of High Speed Embedded – An Algorithmic Procedure', *WSEAS Transactions on Computers*, vol. 10, issue no. 1, pp. 16 – 20, 2011

Author (Short Biography)

Afaq Ahmad belongs to department of Electrical and Computer Engineering department at Sultan Qaboos University, Sultanate of Oman. He holds B.Sc. Eng., M.Sc. Eng., DLLR and Ph.D. degrees. Ahmad did his PhD from IIT Roorkee, India in 1990. Before joining Sultan Qaboos University, Dr. Ahmad was Associate Professor at Aligarh Muslim University, India. Prior to starting carrier at Aligarh, he also worked as consultant engineer with Light & Co., lecturer with REC Srinagar and senior research fellow with CSIR, India.

Dr. Ahmad is Fellow member of IETE (India), senior member of IEEE Computer Society (USA) and life member of SSI (India), senior member IACSIT, member IAENG and WSEAS; He has published over 100 technical papers. At present he is associated as editors and reviewers of many reputed journals. He has delivered many keynote, invited addresses, extension lectures, organized conferences, short courses, and conducted tutorials at various universities of globally repute. He chaired many technical sessions of international conferences, workshops, symposiums, seminars, and short courses. He has undertaken and satisfactorily completed many highly reputed and challenging consultancy and project works. His research interests are: fault diagnosis and digital system testing, data security, graph theoretic approach, microprocessor and microcontroller based systems, advanced logic design and interfacing using Verilog – HDL, system reliability, fault tolerance, algorithm design, test and computer programming.

Dr. Ahmad's field of specialization is VLSI testing, fault-tolerant computing, data security and error detecting and coding.