Indian Journal of Science and Technology

On Properties of PN Sequences Generated by LFSR – a Generalized Study and Simulation Modeling

Afaq Ahmad*, Sayyid Samir Al-Busaidi and Mufeed Juma Al-Musharafi

Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University,
P. O. Box 33, Postal Code 123; Muscat, Sultanate of Oman; afaq@squ.edu.om, albusaid@squ.edu.om,
mufeed03@hotmail.com

Abstract

This paper presents a study and developed simulation models for testing properties of pseudo-noise sequences. A generalized approach is considered while presenting the study of properties of pseudo-noise sequences. On the other hand MATLAB SIMULINK which a commonly available computing platform is used to develop the simulation models for testing the properties of pseudo-noise sequences.

Keywords: LFSR, PN Sequence, MATLAB, Autocorrelation, Run Length.

1. Introduction

Pseudo-Noise (PN) sequences whose terms depend in a simple manner on their predecessors are of great importance for a variety of other applications. Such sequences are easily generated by recursive procedures and hence PN sequences have an advantageous feature from the computational viewpoint, and they tend to have useful structural properties. Due to only these structural properties, PN sequences have enormous applications like Direct Sequence Spread Spectrum (DSSS), Built-in Self-Test (BIST), Decryption–Encryption System (DES) error detection and many more [1–12].

In these applications the systems use the basic hardware named Linear Feedback Shift Register (LFSR) to generate Pseudo-Noise (PN) sequence [1–13]. As shown in Figure 1, an LFSR is made up of two parts. These parts are a shift register and a feedback function. The shift register is a chain sequence of n-bits of D – type of Flip-Flops (FFs). Each time a new bit is needed to load the first bit (D-FF$_1$) of the chains of D – FFs. The all others of the bits in the shift register are shifted one bit to the right. The feedback function is simply the Exclusive-OR (EOR) operation logic of certain bits of the register. The list of those bits which are involved in EOR operation logic is called a feedback taps (C_0 C_1 C_2 ….. C_i …. C_{n-1} C_n). The new left most bit's state (first bit of D – flip-flop, D-FF$_1$) is computed as a function of the existing feedback taps of LFSR. The output of the feedback shift register is one bit at each clock, often the most significant bit a clock before. The period p of a shift register is the length of the output sequence before it starts repeating [1–13].

Linear feedback shift registers make extremely good PN sequence generators of desired period length. A maximal length of $p = 2^n-1$ can be generated through an n-bit LFSR. When the flip-flops are loaded with a seed (initial condition) value (any thing except all 0s, which would cause the LFSR to produce all 0 patterns) and when the LFSR is locked, it will generate a PN sequence of 1s and 0s. Note that the only signal necessary to generate PN sequence is the clock and initial loading of LFSR [1–13].

*Corresponding author:
Afaq Ahmad (afaq@squ.edu.om)

Table 1 shows the patterns produced by the LFSR shown in Figure 2, assuming that the pattern of 11111 was used as an initial loading and feedback taps are taken from 3rd and 5th bits of the FFs.

The generated PN sequences outputted from FF_5, FF_4, FF_3, FF_2, and FF1 are:

PN_5 = [1111100011011101010000100101100],
PN_4 = [1111000110111010100001001011001],
PN_3 = [1110001101110101000010010110011],
PN_2 = [1100011011101010000100101100111], and
PN_1 = [1000110111010100001001011001111] respectively.

Each PN sequences has the same period p = 31. Also, it is notable all the PN sequences are observing the same properties for example total number of 1s and total number of 0s, groups of 1s and 0s and like many other properties. These properties have been studied by many researchers [1–26]. Through this paper we want to present a tool along with study of some properties. The developed tool is based on MATLAB SIMULINK [27–32] and easy to modify.

2. Study of Properties of PN Sequences

The properties of PN sequences are described in many ways [1–31]. However, in this section we describe the properties of PN sequences in a unique and generalized way.

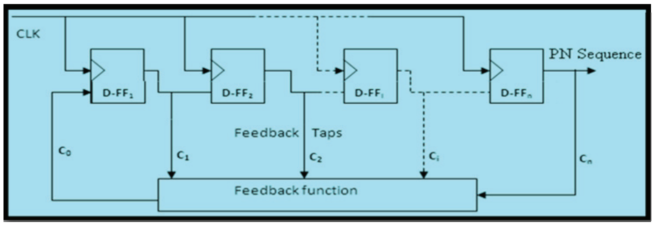

Figure 1. An n-bit LFSR structure.

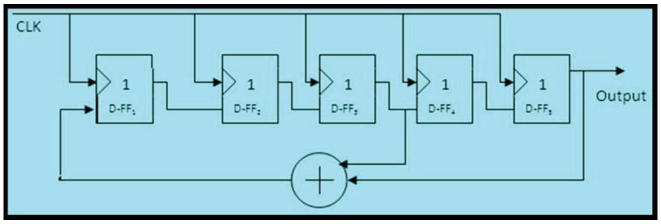

Figure 2. A 5-bit LFSR structure.

Property 1: P1

In every period (p = 2n−1) of PN sequence generated by an n-bit LFSR, the sequence will contain the total number of 1s equal to 2n−1.

Property 2: P2

In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence will contain the total number of 0s equal to 2^{n-1}-1.

Property 3: P3

In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence has an occurrence of n number of 1s in succession.

Table 1. The patterns of pn sequences produced by the lfsr of Figure 2

Clock	FF_1	FF_2	FF_3	FF_4	FF_5	Comment
1	1	1	1	1	1	Initial Loading
2	0	1	1	1	1	
3	0	0	1	1	1	
4	0	0	0	1	1	
5	1	0	0	0	1	
6	1	1	0	0	0	
7	0	1	1	0	0	
8	1	0	1	1	0	
9	1	1	0	1	1	
10	1	1	1	0	1	
11	0	1	1	1	0	
12	1	0	1	1	1	
13	0	1	0	1	1	
14	1	0	1	0	1	
15	0	1	0	1	0	
16	0	0	1	0	1	
17	0	0	0	1	0	
18	0	0	0	0	1	
19	1	0	0	0	0	
20	0	1	0	0	0	
21	0	0	1	0	0	
22	1	0	0	1	0	
23	0	1	0	0	1	
24	1	0	1	0	0	
25	1	1	0	1	0	
26	0	1	1	0	1	
27	0	0	1	1	0	
28	1	0	0	1	1	
29	1	1	0	0	1	
30	1	1	1	0	0	
31	1	1	1	1	0	
32	1	1	1	1	1	Starts repeating

Property 4: P4

In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence does not have any occurrence of total number of (n) 0s in succession.

Property 5: P5

In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence does not have any occurrence of total number of (n−1) 1s in succession.

Property 6: P6

In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence has an occurrence of total number of (n−1) 0s in succession.

Property 7: P7

We define the term 'run' in a general way as a succession of items of the same class. In a period of PN sequence the distribution of sequential occurrences of groups of 1s, and 0s (runs property for $1 \le k \le n-2$), is governed by a rule and we present this in the form of the following theorem.

THEOREM 1. In every period ($p = 2^n-1$) of PN sequence generated by an n-bit LFSR, the sequence will contain 2^{k-1} runs of (n - k - 1) 1s, as well as 0s, for $1 \le k \le n-2$.

We prove our stated properties (P1 to P6) and Theorem 1 by presenting an example as below.

Example 1. Let us consider a 4-bit LFSR initially loaded with 0011 and has feedback taps from 3rd and 4th FFs. The generated PN sequence has its length $p = 2^4-1 = 15$. The sequence generated by this structure of LFSR is, PN4 = [001101011110001].

The Properties P1 – P6 and Theorem 1 can be verified by analysing the generated sequence PN4. For better explanation the analysis result is presented below in the form of the Table 2.

Table 2. The run counts / patterns of 1s and 0s in PN4

Number of Runs	Succession of	Comment
1	4 – 1s	→ P3
0	4 – 0s	→ P4
0	3 – 1s	→ P5
1	3 – 0s	→ P6
1	2 – 1s	→ P7 / Theorem 1
1	2 – 0s	→ P7 / Theorem 1
2	1 – 1s	→ P7 / Theorem 1
2	1 – 0s	→ P7 / Theorem 1
Number of 1s = 8		→ P1
Number of 0s = 7		→ P2

Property 8: P8

It is also interesting to note that the LFSR generates pulses of different frequencies. The study also, reveals that the pulse width and frequency of different pulses has definite relation with the others as shown in Figure 3. Table 3, describes this property for a PN sequence generated by the LFSR of Figure 2. The generated sequence has periodicity $p = 2^n-1$ with assumption that the clock pulse of LFSR has time period T.

Property 9: P9 (The property of auto-correlation)

To study the statistical property PN sequences, it is important to analyse them through their correlation functions. Correlation function of two sequences can be described as the comparison of two sequences to see how much they correspond with one another. Various parameters effect the correlation of two sequences including the length of sequence, phase between the sequences, and clock rate of LFSR. The act of correlating a signal through all variations

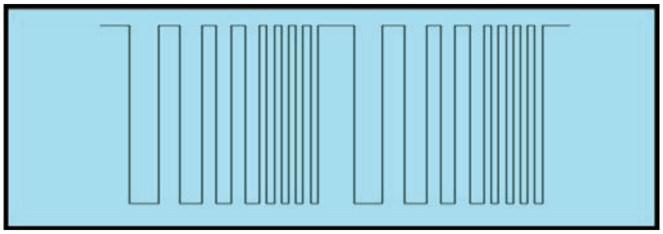

Figure 3. Waveform of the sequence generated by the LFSR of Figure 2.

Table 3. Pulses generated by the LFSR of Figure 2

Number of pulses	Pulse width	Nature of pulse
1	n*T	Active high
1	(n−1) * T	Active low
For $1 \le x \le n-2$; 2^{x-1}	(n−x−1) * T	Active high
For $1 \le x \le n-2$; 2^{x-1}	(n−x−1) * T	Active low

of itself is known as autocorrelation. The autocorrelation $((s_{i\in N}) = S_1, S_2,...S_i.....)$ function, AC (k) of an m-sequence where, N = 1 to 2^n-1 can be given for its kth shift as:

$$AC(k) = \frac{1}{N}\sum_{i=1}^{N} s_i s_{i+k}; \qquad 0 \le k \le N-1 \tag{1}$$

where, s_i is the value of the ith - position of the PN sequence.

THEOREM 2. The autocorrelation function of an m-sequence reaches a maximum of 2^n-1 at zero shifts. Whereas, for other shifts ($0 \le k \le N-2$) its value will be equal to –1.

It can be visualized through Figure 4 that the peaks of autocorrelation values (AC (k); Equation 1) are as 31 at zero shifts, and for other shifts the values are –1, hence, verifying the Theorem 2. Further, it can be seen that it repeats in each cycle of the generated m-sequence. The result is satisfying the autocorrelation property of the m-sequence as given in Equation 1.

3. Simulation Models

We used the commonly available computing facility platform to simulate the models of our studies. MATLAB SIMULINK [32] is used to provide the models. Below are the classifications of all the developed models.

3.1 Generation of PN Sequence and Waveform Observation

Figure 5 depicts the simulation model in MATLAB SIMULINK for the LFSR of Figure 2. The output file of PN sequence in binary form is as given FF_5 in Table 1, whereas the oscilloscope waveform is as shown in Figure 3.

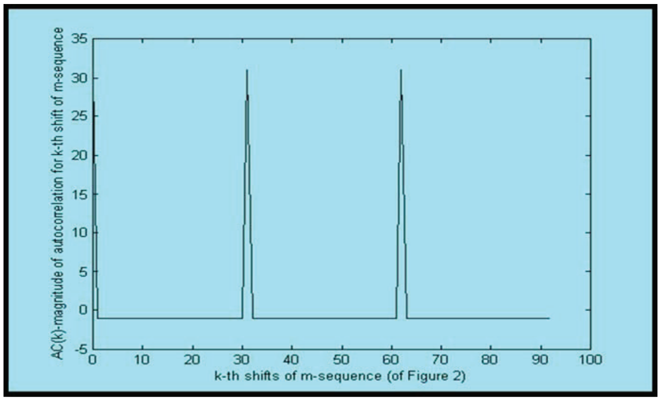

Figure 4. Autocorrelation property of the sequence generated by the LFSR of Figure 2.

3.2 Counting Number of 1s and 0s

To test a PN sequence for its property of sequence length and number of 1s and 0s a simulated model is developed as given Figure 6. In the model when the binary input sequence is applied to the Relational Operator, it checks the presence of ones and accordingly updates the Memory (counter). Input 2 of Switch functions as a controller, if it is TRUE, then it let's to pass the output of the Sum 1 to Memory through input 1 of Switch. ELSE, Input 3 of Switch becomes operative to feed the Memory. Since the Digital Clock is responsible for counting the total number of bits of the sequence, thus, the subtraction of the present status of the Memory from the Clock will exactly represent the presence of the number of zeros in the sequence.

3.3 Testing Run Lengths

The run lengths of generated PN sequence can be tested with the simulated model shown in Figure 7. The logic of this simulation model is that the first process in the simulation of run length is to keep track of transitions and then to count the numbers in each of the transitions. In

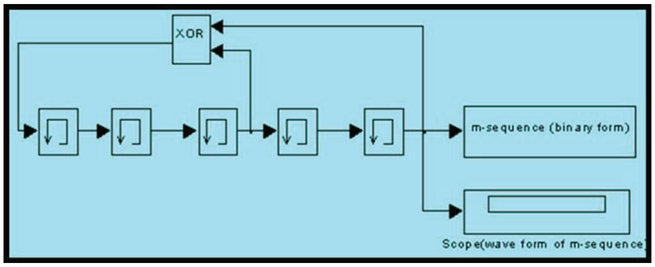

Figure 5. MATLAB SIMULINK model for the LFSR of Figure 2

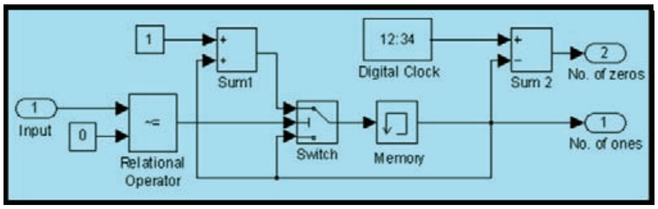

Figure 6. MATLAB SIMULINK model for counting 1s and 0s of PN sequence.

the figure, Subsystem block receives the binary sequence and produces two outputs. Output 1 monitors the run length property of the input sequence. The detailed simulation model for block Output 1 is described in Figure 8.

In the Figure 8, the difference of previous and present input bits is being fed to Relational operator 1. The operator then checks whether transition between the bits exists or not. The each difference value 0, 1 and -1 reflects no transition, transition from 1→ 0, and transition from 0 → 1 respectively. The Memory keeps tracks of number of transitions, which receives the data from Sum 1. Switch 1 and Switch 2 are feeding the block Sum 1. The purpose of controller (Relational operator 2) of Switch 2 is just to initialize the Memory. The function of Sum 2 is to keep Memory updated with the counts of the changed transitions. To demonstrate the discussed functional procedure of the different simulated blocks, an example is presented as below:

Example 2. Let the PN input sequence is 10100111. So that the resulted number of transitions for the given sequence

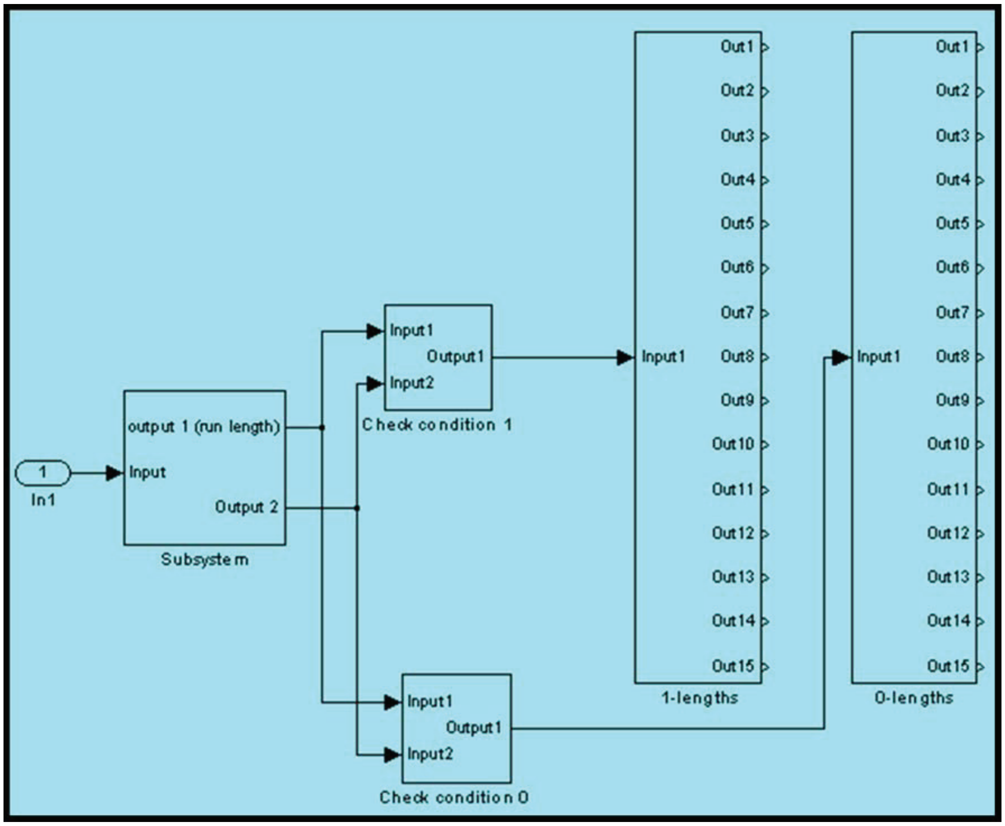

Figure 7. Simulation model for run length test.

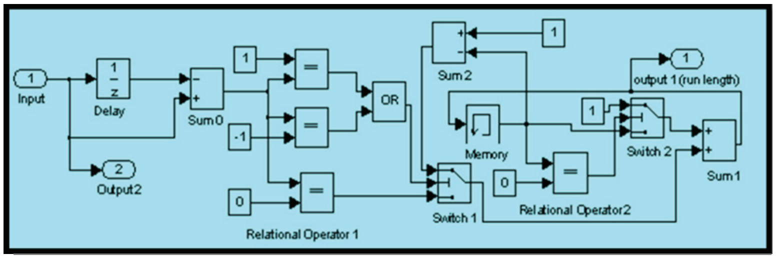

Figure 8. Simulation model for subsystem (output 1) of Figure 7.

is 1112123. The status of all the blocks of Figure 8 can be visualized in Table 4.

The above-obtained result through the Output 1 of the Subsystem block of Figure 7 is simultaneously fed to the inputs of the blocks of Check condition 1 and Check condition 0. The realized model of Check condition 1 and Check condition 0 are given in Figs. 9 and 10 respectively. The objectives of realizing these two blocks are only to differentiate and/or to determine the sequential lengths of 1s and 0s in the transition counts. The Table 5 explains the results of the outputs of the simulation models of Figures 9 and 10 for the above obtained transition counts 1112123 (from Example 2).

The outputs of the Check condition 1 and Check condition 0 are now inputted to 1-lengths and 0-lengths blocks, respectively (see Figure 7). The purpose of simulating these blocks is due to the reason that we are interested in counting and displaying the runs of 1s and 0s. Out n (1 ≤ n ≤ 15) describes the number of n 1s / 0s. The Figure 11 demonstrates the simulation model of block Out 1.

3.4 Autocorrelation Test

The autocorrelation of the PN sequence can be computed using the scripted MATLAB program given below.

Table 4. Status of different blocks of Figure 8

Clock states	1	2	3	4	5	6	7	8
Inputs to Sum 0	1,0	0,1	1,0	0,0	0,1	1,1	1,1	×
Output from Sum 0	1	-1	1	0	-1	0	0	×
Status of controller of Switch 1	1	1	1	0	1	0	0	×
Output from Sum 2	1	0	0	0	-1	0	-1	×
Output from Switch 1 (either from input 1 or input 3)	1	0	0	1	-1	1	1	×
Memory counter value	0	1	1	1	2	1	2	3
Output from Relational operator 2	1	0	0	0	0	0	0	×
Output from Switch 2 (either from input 1 or input 3)	1	1	1	1	2	1	2	×
Output from Sum 1	1	1	1	2	1	2	3	×

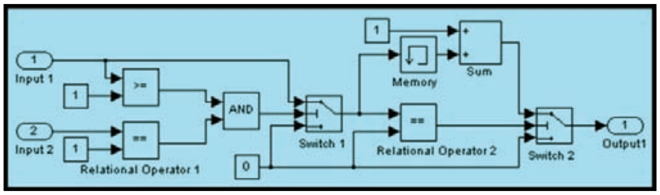

Figure 9. Simulation model for check condition 1 block of Figure 7.

3.5 Program (MATLAB Code)

```
function auto=auto_corr(seq)
m=length(seq1);
seq=(seq1-0.5)*2;
for i=1:m
   auto(i)=seq*seq';
   r=seq(1);
   seq(1:m-1)=seq(2:m);
   seq(m)=r;
end
```

4. Conclusions

We run the developed MATLAB-SIMULINK models for various generated PN sequences. The results for the counts of number of 1s and number of 0s as well as run lengths are checked. The waveforms and autocorrelation of PN sequences are studied through the developed models.

Thus, based on a systematic and procedural study of the theory of PN sequences we developed a tool to facilitate the practicing engineers to either generate PN sequences for its application or to test PN sequences applied or both. Since the security is a vital issue in this age of information technology, and finally, it seems that the security responsibility has to come in any form on the shoulders of practicing engineers of all fields to avoid the litigations.

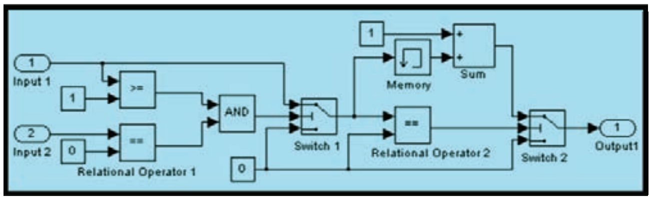

Figure 10. Simulation model for check condition 0 block of Figure 7.

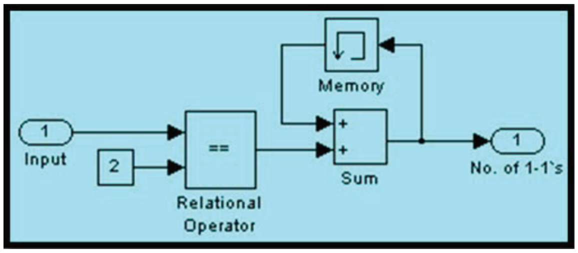

Figure 11. Simulation model for out 1 blocks of Figures 9 and 10.

Table 5. Results of the outputs of Figures 9 and 10

Input 1	**1**	**1**	**1**	**2**	**1**	**2**	**3**
Outputs from Check condition 1 and Check condition 0; in terms of Runs of 1s and 0s							
Check condition 1	1	0	1	0		3	
Check condition 0	0	1	0	2		0	

5. Acknowledgement

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.

6. References

1. Ahmad A (2012). Better PN generators for CDMA application – a Verilog-HDL implementation approach, International Journal of Information Engineering (IJIE), vol 2(1), 6–11.
2. Ahmad A, Al-Abri D et al. (2012). Adding pseudo-random test sequence generator in the test simulator for DFT approach, Journal of Computer Technology and Applications (JCTA), vol 3(7), 463–470.
3. Ahmad A, and Hayat L (2011). Selection of polynomials for cyclic redundancy check for the use of high speed embedded – an algorithmic procedure, Transactions on Computers (WSEAS), vol 10(1), 16–20.
4. Ahmad A (2010). 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, 1–12.
5. Ahmad A, and Al-Balushi J (2009). How to design an effective Serial Input Shift Register (SISR) for data compression process of built-in self-test methodology, Proceedings 4th International Design and Test Workshop (IDT'09) held at King Abdul Aziz City for Science and Technology (KAASAT), Riyadh, Saudi Arabia. 372–379.
6. Al-Naamany A M, and Ahmad A (2003). Development of a strong stream ciphering technique using non-linear fuzzy logic selector, Mobile and Wireless Communications, IFIP — The International Federation for Information Processing, vol 106, 199–206.
7. Ahmad A, Al-Musharafi M J et al. (2002). Design and study of a strong stream crypto-system model for e-commerce, Proceeding ICCC '02 Proceedings of the 15th International Conference on Computer Communication, vol 1, 619–630.
8. Jamil T, and Ahmad A (2002). An investigation in to the application of linear feedback shift registers for steganography, Proceedings IEEE SoutheastCon2002, Columbia, SC, USA, 239–244.
9. Ahmad A, Al-Musharafi M J et al. (2001). An NLFSR based sequence generator for stream ciphers, Proceedings (SETA'01) - An International Conference on Sequences & Their Applications, Norway (Bergen), 11–13.
10. Golic J D (2000). Cryptanalysis of three mutually clock-controlled stop/go shift registers, IEEE Transactions on Information Technology, vol 46, No. 3, 1081–1090.
11. Ahmad A (1997). Achievement of higher testability goals through the modification of shift register in LFSR based testing, International Journal of Electronics (UK), vol 82, No. 3, 249–260.
12. Chen H W, Aine C J E et al. (1996). Nonlinear analysis of biological systems using short m-sequences and sparse-simulation techniques, Annals of Biomedical Engineering, vol 24, 513–536.
13. Golomb S W (1982). Shift Register Sequences, Aegean Park Press, Revised Edition.
14. Ahmad A, Nanda N K et al. (1990). Are primitive polynomials always best in signature analysis?, IEEE Design & Test of Computers (USA), vol 7, No.4, 36–38.
15. Blum L, Blum M et al. (1986). A simple unpredictable pseudo-random number generator, SIAM Journal of Computing, vol 15, No. 2, 364–383.
16. Ahmad A, Nanda N K et al. (1988). A critical role of primitive polynomials in an LFSR based testing technique, IEE Electronics Letters (UK), vol 24(15), 953–955.
17. Ahmad A, Al-Busaidi S S et al. (2013). Study on cyclic cross-correlation behavior of maximal length pseudo-random binary sequences, Indian Journal of Industrial and Applied Mathematics (Taylor & Francis), vol 4(1), 33–43.
18. Ahmad A, and Al-Abri D (2012). Design of a pseudo-random binary code generator via a developed simulation model, International Journal on Information Technology (ACEEE - Journal), vol 2(1), 33–36.
19. Ahmad A (2011). Investigation of some quite interesting divisibility situations in a signature analyzer implementation, Transactions on Circuits and Systems (WSEAS), vol 10 (9), 299–308.
20. Ahmad A (1998). An algorithmic generation of sparse primitive polynomials of order n, International Wireless and Telecommunication Symposium (IWATS'98), Shah Alam (Malaysia).
21. Ahmad A, and Elabdalla A M (1997). 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 (Elsevier), vol 23(1), 33–39.
22. Ahmad A, and Al-Maashri A (2008). Investigating some special sequence length generated through an external exclusive-NOR type LFSRs, International Journal Electrical and Computer Engineering, vol 34(1), 270–280.

23. Ahmad A (2003). Realization of a stream cipher with better security and higher reliability goals, 2nd International Conference on Quality, Reliability and Information Technology - Trends and Future Directions, 18–21, 77–78.
24. Ahmad A, Al-Busaidi S et al. (2003). Measurement techniques of LFSR sequences, Proceedings International Symposium on Wireless Systems and Networks (ISWSN'03), King Fahad University of Petroleum - Dahran, Kingdom of Saudi Arabia, 1–5.
25. Ahmad A, Nanda N K et al. (1989). The use of irreducible characteristic polynomials in an LFSR based testing of digital circuits, Proceedings of 4th IEEE International Conference of Region 10 (TENCON-89), 494–496.
26. Ahmad A, Ahmed Al-Mashari et al. (2009). On locking conditions in M-sequence generators for the use in digital watermarking, Proceedings International Conference on Methods and Models in Computer Science (ICM2CS09) held at School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India, 111–115.
27. Ahmad A, Al-Musharafi M J et al. (2001). Study and implementation of properties of m-sequences in MATLAB-SIMULINK – A pass / fail test tool for designs of random generators, Proceedings IEEE / IEE International Conference on Communication, Computer and Power (ICCCP'01), Oman, 191–196.
28. Ahmad A, Al-Musharafi M J et al. (2002). Study and implementation of properties of m-sequences in MATLAB-SIMULINK – A pass / fail test tool for designs of random generators, Journal of Scientific Research – Science and Technology, vol 7 (part 1), 147–156.
29. Al-Lawati A, and Ahmad A (2004). Realization of a simplified controllability computation procedure – a MATL AB-SIMULINK based tool, Sultan Qaboos University Journal for Scientific Research - Science and Technology, Oman, vol 8, 131–143.
30. Ahmad A, and Ruelens D (2013). Development of digital logic design teaching tool using MATLAB & SIMULINK, IEEE Technology and Engineering Education (ITEE), vol 8, No. 1, 7–12.
31. Ahmad A, Ruelens D et al. (2013). Development of verification tool for minimal Boolean equation, IEEE Technology and Engineering Education (ITEE), vol 8, No. 3.
32. MATLAB: Available from: http://www.mathworks.com/

7. Author Biography

AFAQ AHMAD, a senior member of IEEE and IEEE Computer Society, received his Ph. D in computer engineering from Indian Institute of Technology, Roorkee in 1990. He obtained M.Sc. degree in electronics engineering in 1977, B.Sc. degree in electrical engineering in 1974 from Aligarh Muslim University, India. He also earned a post graduate diploma in industrial management 1n 1980. Currently, he is working at Sultan Qaboos University in Oman. Before joining Sultan Qaboos University, he was reader (associate professor at Aligarh Muslim University in 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 council of scientific and industrial research (CSIR) of India. Dr. Ahmad a recipient of various student scholarships, award and recognitions has authored more than 110 scientific papers, book chapters, numerous technical reports, and manuals. He received best scientific paper awards. His field of specialization is VLSI testing, algorithm design and testing, fault-tolerant computing, data security, coding and its commercial applications and development low cost engineering educational tools. He has undertaken and satisfactorily completed many highly reputed and challenging consultancies and project works. He serves as editor, associate editor and member international advisory boards for many worlds' reputed journals. He is honored and continued to serve as program chairs, technical chairs and tutorial chairs for many IEEE sponsored annually organized conferences. He chaired many technical sessions, meetings and panel discussions of international conferences, symposia and meetings. He conducted many workshops and short courses. He has delivered many invited talks and keynote addresses on current issues in various areas of importance. He has over thirty five years of professional experience with universities and industries. He proved himself as a reputed and excellent instructor who keeps the pace in the changes of topics, courses and techniques of learning processes. He has credit of developing the curricula and programs for various educational institutions. He has extensive administrative experiences of various levels required in operation and management of universities and institutes. He is Fellow member of IETE, senior member of IEEE and IEEE Computer Society, life member of SSI, senior member of IACSIT, member of IAENG and honorary member of Enformatika and WSEAS societies.