Increasing Randomness for Pseudo Random Number Generator

Avinash Mishra[1], Dr. Yuan Ma [2]

[1] *Graduate Student, Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada*
[2] *Professor, Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada*

ABSTRACT

Pseudo-random number generation (PRNG) technique is widely used for simulation and cryptographic applications. For the pseudo-random number generation, a linear feedback shift register (LFSR) is being used, due to the ease of construction from a simple electronic circuit. In this paper, a detailed analysis of LFSR and techniques to generate random numbers has been presented. Further, the challenges with PRNG are mentioned and a technique, i.e cryptographic pseudorandom number generator (CPRNG) is described to overcome the challenges and to increase the randomness to make the system secure and robust.

Keyword : - *LFSR, PRNG, CPRNG*

1. Introduction

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. It is a shift register whose input bit is driven by the exclusive-or (xor) or exclusive-nor (xnor) of some bits of the overall shift register value. In the diagram below [1], we consider a linear feedback shift register of length four.

The output from each shift registers ranges from x_1 to x_4.

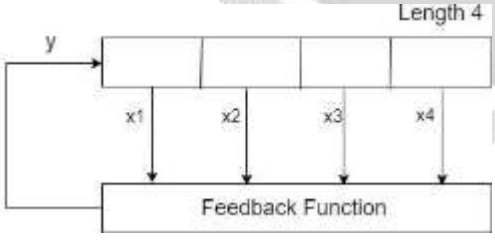

Figure 1 Basic block diagram of LFSR [1]

From the above diagram, the feedback function is expressed as:

$$y = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4$$

Where c is either 0 or 1. When 1, the output is tapped. Also, the sum in the above equation is obtained through the xor or xnor operation.

2. LFSR POLYNOMIALS

A. Feedback Polynomial

The feedback polynomial is expressed in terms of delay for each shift register. Here, the delay is expressed in terms of x^0 tox^4 which is shown in the figure below.

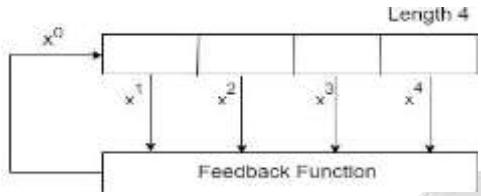

Figure 2 LFSR with delay in each shift register [2]

From the above diagram, the feedback polynomial is expressed
$f(x) = 1+ c_1x^1 + c_2x^2 + c_3x^3 + c_4x^4$

Here, for the n-bit shift register, the degree of the polynomial is n

B. Primitive Polynomial

A polynomial that produces maximum combinations, i.e., (2^n-1)sequence is called a primitive polynomial.

Table 1 is an example of the primitive polynomial for the mentioned degree.

Table 1 Polynomials for various degrees of LFSR

Degree	Polynomial
2,3,4,6,7	x^n+x+1
5	x^5+x^2+1
8	$x^8+x^6+x^5+x+1$
9	x^9+x^4+1
10	$x^{10}+x^3+1$

From the Table 1, some necessary conditions to be a primitivepolynomial can be deduced:

I. Number of taps is even
II. Taps numbers are co-prime
III. If tap sequence for n-bit LFSR generating primitive polynomial is n,m,l,k,….,0, then the tap sequence n-n,n-m, n-l, n-k,..,n-0, i.e. 0, n-m, n-l, n-k, ….., n will also give primitive polynomial. For example, for the fifth degree, a primitive polynomial expression is
$x^5 + x^2+1$
this gives, $x^{5-5} + x^{5-2} + X^{5-0}$
$= X^5 + X^3 +1$ is also a primitive polynomial

3. PSEUDO-RANDOM NUMBER GENERATION

A. Through Standard LFSR

Standard LFSR consists of external feedback [3]. Here, the pseudorandom sequence is generated through the shift register whose serial input is the exclusive-or of certain bits.

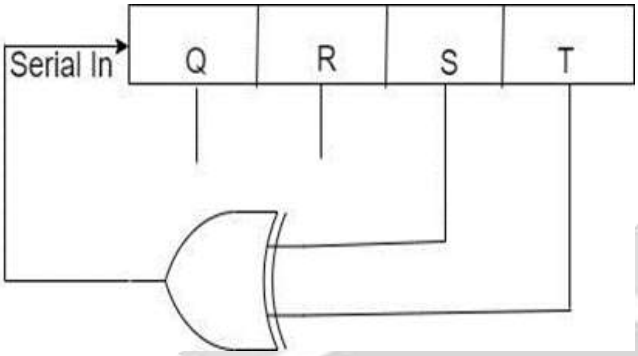

Figure 3 Standard LFSR [3]

Here, Q, R, S and T are the shift register and the input to the shift register is the XOR of the value of S and T. Further,the above diagram can be reduced as:

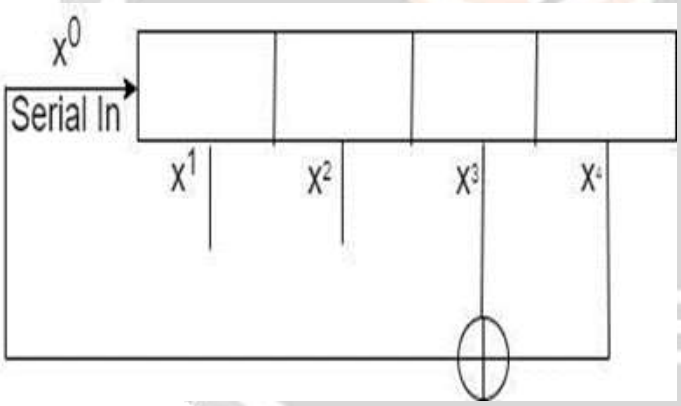

Figure 4 Simplified standard LFSR with serial input

The above circuit can be expressed as $f(x) = x^4 + x^3 + 1$ and it takes 15 clock cycles to repeat the initial input. The point to be noted for the pseudo-random generation is that the initialization can be done by all the bits as zeros. Because all zeros will lock the state and there will be no change in state.

Table 2 specifies the random number generation in each clockcycle

Table 2 Serial input for Standard LFSR

Q	R	S	T	Serial In = S XOR T	Clock Cycle
1	1	1	1	0	1
0	1	1	1	0	2
0	0	1	1	0	3
0	0	0	1	1	4
1	0	0	0	0	5
0	1	0	0	0	6
0	0	1	0	1	7
1	0	0	1	1	8
1	1	0	0	0	9
0	1	1	0	1	10
1	0	1	1	0	11
0	1	0	1	1	12
1	0	1	0	1	13
1	1	0	1	1	14
1	1	1	0	1	15
1	1	1	1	0	16(repeat)

B. Through Modular LFSR:

It consists of internal feedback and the internal feedback representation of the circuit [4] is shown below:

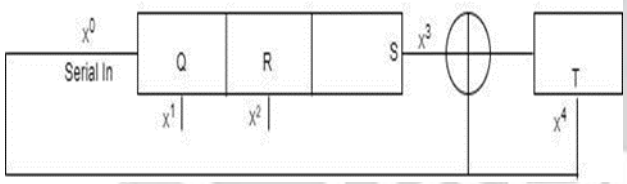

Figure 5 Modular LFSR [4]

The above circuit can be represented by the followingpolynomial equation:

$f(x) = x^4 + x^3 + 1$

Similar to the external feedback structure, it will take 15 clockcycle to repeat itself which can be demonstrated by the table below:

Table 3 Serial input for Modular LFSR

Q	R	S	T	Serial In = T	Input to T FF	Clock Cycle
1	1	1	1	1	0	1
1	1	1	0	0	1	2
0	1	1	1	1	0	3
1	0	1	0	0	1	4
0	1	0	1	1	1	5
1	0	1	1	1	0	6
1	1	0	0	0	0	7
0	1	1	0	0	1	8
0	0	1	1	1	0	9
1	0	0	0	0	0	10
0	1	0	0	0	0	11
0	0	1	0	0	1	12
0	0	0	1	1	1	13
1	0	0	1	1	1	14
1	1	0	1	1	1	15
1	1	1	1	1	1	repeat

4. CHALLENGES WITH PRNG

A PRNG is typically composed of a seed, an output function producing some random bits from the seed, and a feedback function that iteratively transforms the seed. Further, it can beseen that a PRNG is necessarily periodic, i.e., the sequence it produces will repeat itself after a certain period. Therefore, pseudo-random number generators cannot be used for data encryption as the random numbers generated by them are predictable. For data encryption, the numbers should be unpredictable. Also, to be used for cryptography,it is necessary to pass the statistical test. Since any statisticaldefect can be used by an attacker to gain information about the PRNG's future output.

5. PROPOSED SOLUTION

A.Cryptographic Pseudo random number Generator

Cryptographic pseudorandom number generator (CPRNG) is a pseudorandom number generator with properties that makesit suitable for use in cryptography. There are two majors advantages of CPRNG:

- ➢ It passes statistical randomness test.
- ➢ It holdup well under serious attack, even when partof initial or running state becomes available to the attacker.

B. Implementation

CPRNG are implemented by aggregating the raw entropy(unpredictable input) for the system [5]. The source of entropy can be:

- ➢ Clock jitter collected during boot
- ➢ Hardware interrupts
- ➢ Random Instructions

In this technique, the n-bit cryptographic random numbers areobtained by gathering "physical events" which should be unpredictable. Usually for this, timing is used, i.e., the CPU has a cycle counter which is updated several billion times per second, and some events occur with an inevitable amount of jitter. Now, once we have n bits, we use a Pseudo-Random Number Generator (PRNG) to make out as many bits as necessary. A PRNG is

said to be cryptographically secure if assuming that it operates over a wide enough n-bit, its output is computationally indistinguishable from uniformly random. To observe increase in randomness with the increase in bits, below the comparison between the simulation of 4 bit and 16 bit LFSR is being done

6. COMPARISON

TO observe the randomness, 4-bit LFSR data is simulated using Xilinx Vivado and the result showing randomness is shown below:

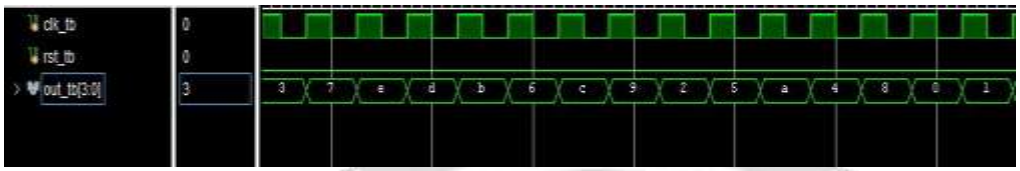

Figure 6 Representing randomness for 4 Bit LFSR

Further, the 16-bit LFSR is simulated, and the randomness is shown below in the simulation

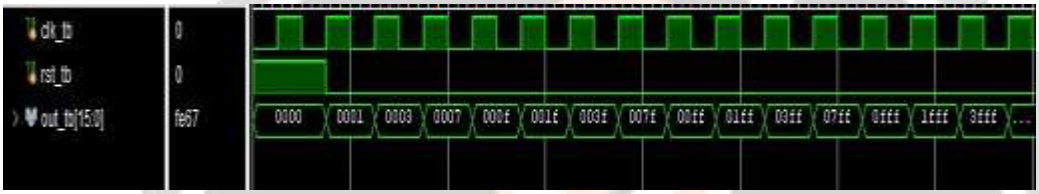

Figure 7 Representing randomness for 16 Bit LFSR

7. CONCLUSION

The characteristics of PRNG can be analyzed from the simulation result shown above. It is seen that the randomness increases with the increase in number of bits. The randomness is more for 16-bit LFSR as compared to 4-bit LFSR. Hence, toincrease the randomness, the cryptographic random number generation technique, which includes various ways to increasethe number of possible random numbers is used. And such technique is recommendable to be used to make the system secure and free from external attacks.

REFERENCES

[1] Wikipedia contributors, "Pseudorandom generator," *Wikipedia, The Free Encyclopedia*, 19-Apr-2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Pseudorandom_generator&oldid=1083577567

[2] Xilinx pseudo random number generator, www.xilinx.com December 2001.

[3] B.Dharma Teja, V.Swetha, D.Lokesh,lokesh. K.V.R.L.Prasad, "Design and Analysis of a 128 bit Linear Feedback Shift Register Using VHDL",International Jornal of Advances in Engineering & Technology(IJAET), Feb 2016, Issue 2, ISSN:2350-0328

[4] Pseudorandom number generation means pseudosecurity, synopsys.com/blogs/software-security/pseudorandom-number-generation August 2016

[5] Random numbers generation, support.apple.com/en-ie/guide/security/seca0c73a75b/web