Random Number Generation using LFSR and Cellular Automata

Shivasari Sanjay[1] Perla Thirupathaiah[2]
[1]Student [2]Assistant Professor
[1,2]Department of Electronics and Communication
[1,2]Sagar Institute of Technology (SITECH), Flame of Forest, Urella – Chevella Road, Chevella, RR.Dist.

Abstract— There are several methods possible to generate random numbers. Most of the true random number generators are based on natural phenomena, which are supposed to be random like variation of instantaneous amplitude (voltage / current) of thermal noise. There are several methods of generation of number sequences which appear to be random, but deep analysis brings out their predictability characteristics. They are referred to as Pseudo random number generators. There are program (or) software based random number generators using a complicated transfer function. There are also hardware based random number generators (or) random vector generators which find extensive applications in VLSI testing. One of the methods that is adopted is based on using LFSR (Linear Feedback Shift Register) with appropriate feedback polynomials. In LFSRs the feedback is generally from output to the intermediate stages. And another method similar to the process that is used in LFSR, in which instead of providing the feedback from output to individual nodes, the feedback is provided from the neighboring nodes. This is called Cellular Automata. It is believed that the randomness of the sequences generated are very close to the natural random phenomena and there is considerable interest in generating random numbers / vectors for several applications in VLSI testing (or) communications etc. I propose to study LFSR as a random number generator in the first instance with an aim to generate test vectors, which are very close to truly random phenomena. Thereafter, I would investigate the characteristics of the sequences generated by different Cellular Automata and compare them between themselves and also with the sequences generated by LFSR. Most of the work will be carried out through simulation.

Key words: LFSR, Cellular Automata, Random Number Generation

I. INTRODUCTION

In the nature one encounters several phenomenon which cannot be predicted exactly like the occurrence of storms or the intensity of storms to some extent, such events are generally unpredictable and they are referred to as random events. Another example of random process is the electrical noise that is generated in electronic components.

These phenomenon's are described using statistical methods like the probability of occurrence of events or average values associate with these events etc. This type of statistical prediction is helpful in planning of operations based on these events. When one takes larger sample of random events, one may be tempted to check how much unpredictable are these events.

For this purpose one tries to resort to statistical measure like correlation between parameters of adjacent events, the probability distribution of the parameters, frequency of occurrence of certain parameters etc. These measures help in evaluating one the randomness of the samples or for utilizing these events for specific applications. There are certain human activities which are closely related to the randomness of some gambling, stock market fluctuations, in cryptography.

II. SCOPE OF THE PROJECT

For doing this project we need to study about linear feedback shift registers and basic principles involved in cellular automata. And also we need to study about tests that are carried out for checking the randomness of generated sequences. The generation of the random number Sequences are using verilog code in Cadence tool. And the tests for randomness are conducted in Mat lab for those generated sequences.

Linear Feedback Shift Registers (LFSRs) and Cellular Automata (CA) are commonly used in the implementation of pseudo-random number generators. However, these designs typically cannot produce high quality random numbers due to adjacent bit correlations and the appearance of repetitive structures in the bit sequences. Expected that by using typical design of Cellular Automata we need to generate the random number sequences which lead to more randomness.

III. RANDOM NUMBER GENERATOR

There are two methods to generate the Random numbers, they are:
1)Software technique (Computer programs)
2)Hardware technique (Circuits)

IV. SOFTWARE TECHNIQUES

One of the most common pseudo random number generation is the linear congruential generator, which uses the recurrence to generate numbers, where a, b and m are large integers, and Xn+1 is the next in X as a series of pseudo-random numbers. $Xn+1= (a Xn + b) \bmod m$

The maximum number of numbers the formula can produce is the modulus, m. To avoid certain non-random properties of a single linear congruential generator, more than two but not many such random number generators with slightly different values of the multiplier co-efficient, a can be used in parallel, with a "master" random number generator that selects from among the several different generators.

V. HARDWARE TECHNIQUES

These are based on shift registers with complex feedback operations. Broadly two types of registers are dealt with:

1) LFSR (Linear Feedback Shift Register).
2) A register in which local feedback is adopted known as Cellular Automata (CA)

VI. CELLULAR AUTOMATA (CA)

Cellular Automata evolves in steps and the value of a node depends on the value of its neighbors. A Cellular Automata (CA) consists of a collection of cells/nodes formed by flip-flops which are logically related to their nearest neighbors using XOR gates. When the value of a node is determined only by two neighboring cells the CA is known as one-dimensional linear CA (for the rest of the text one-dimensional linear CA is referred as a CA). The logical relations which relate a node to its neighbors are known as rules and they define the characteristics of a CA. There are many rules which can be used to construct a CA register; the typical Cellular Automata (CA) configuration is given in figure 2.3.

The next state x (t+1) of node xi is determined by the current state x(t) of neighboring nodes xi-1 and xi+1 for rule 90 and nodes xi , xi-1 and xi+1 for rule 150. All the nodes of a CA register do not have to be implemented with the same rule; different nodes can employ different rules. The first and the last nodes of a CA register have only one neighbor unlike all other nodes which have two; hence normal rules cannot be applied here. One solution is to assume that the missing neighbor is fixed at logic '0' (null boundary condition). Figure shows the construction of a 4-bit CA register using rules 90 & 150 and null boundary condition.

VII. TESTING

Some tests are involved into a variety of subtests. The 6 tests are:

1) The Frequency (Mono bit) Test,
2) Frequency Test within a Block,
3) The Runs Test,
4) The Runs test within a Block,
5) Correlation test
6) Chi-square goodness-of-fit test

VIII. TEST ANALYSIS OF THE LFSR AND CELLULAR AUTOMATA

n-bit LFSR TESTS	4-bit	4-bit	
		M1 block	M2 block
Frequency Tests	n1=8 n0=7 d=1 (6%)	n1=4 n0=3 d=1 (14%)	n1=4 n0=4 d=0 (0.0%)
Runs TEST	Nruns=9 P=0.9432	Nruns=6 P=0.826	Nruns=4 P=1

n-bit CA TESTS	4-bit	4-bit	
		M1 block	M2 block
Frequency Tests	n1=7 n0=7 d=0 (0.0%)	n1=3 n0=4 d=1 (14%)	n1=4 n0=3 d=0 (14%)
Runs TEST	Nruns=9 P=0.9607	Nruns=5 P=0.9100	Nruns=4 P=0.9843

IX. CONCLUSION

We have generated the random number sequences using Linear Feedback Shift Register and tested that random sequence to check whether they are random or non-random. And also generate the random number sequence using Cellular Automata and check for randomness, so that we compare the randomness of generated sequences using both the methods LFSR and CA. In this project we have found that CA will give more randomness of generated random number sequence.

REFERENCES

[1] Pseudo random number generator; http:// /wik en.wikipedia.orgi/Random_number_generator
[2] S Wolfram, Rondom Sequence Generation by Cellular Automata(CA)www.stephenwlfram.com
[3] Text book 'VLSI test principles and architectures Design for testability' by Laung-Terng Wang, Cheng-Wen wu, Xiaoqing Wen, 2006.
[4] http://www.mathwarks.com/help/toolbox/stats/runstest.html.
[5] "Rondom sequence generation by cellular automata," Advanced in Applied Maths

