Comparative Analysis of High Efficient Random Number Generator

*Shruthi K[1], Prasanna Kumar C[1], Akshatha[2]
[1]*(Electronics & Communication Engineering, Sahyadri College of Engineering& Management, India)*
[1]*(Associate professorElectronics & Communication Engineering, Sahyadri College of Engineering& Management, India)*
[2]*(Assistant professorElectronics & Communication Engineering, Sahyadri College of Engineering& Management, India)*
Corresponding Author: Shruthi K

Abstract: *A detailed comparison among pseudo-random number generators (PRNGs) such as Linear Feedback Shift Register(LFSRs), Linear Congruential Generator (LCGs) and Combination of Linear Feedback Shift Register (LFSR) and Cellular Automata Shift Register(CASR) are presented in this paper. Linear PRNGs; such as LFSRs and LCGs can produce long-period random number sequences. When implemented, linear PRNGs are efficient in throughput rate and hardware cost, but the output random numbers of such generators are predictable due to their linear structure. To avoid the predictable nature of the PRNGs, the non-linear combination generator, i.e Combination of LFSR and CASR is made used in this paper. Also to increase the cycle length modern technique like Combination of LFSR and CASR is used. All three techniques are simulated and synthesized using Verilog on the Xilinx ISE 10.1 and also powers of all techniques are compared.*

Keywords: *RNG, Linear Feedback Shift Register, LCG, Cellular Automata Shift Register, Xilinx*

Date of Submission: 17-07-2017 Date of acceptance: 28-07-2017

I. Introduction

Pseudo-random number sequences, also known either as Pseudo-Noise(PN) sequences, or maximal length binary sequences, are widely used in digital computing and communication applications, such as VLSI testing, coding, cryptography, D-A conversion etc. In a truly random sequence the bit pattern is never repeated. However, such a sequence is very difficult to generate and has little use in practical systems. Many applications demand the data to be appear random but predictable to the user. A pseudo-random sequence fulfills the requirements of randomness, but the entire sequence repeats indefinitely. The linear Congruential generator which can be used to produce such sequences are fast and require minimal memory(typically 32 or 64 bits) to retain state. This makes them valuable for simulating multiple independent streams. Linear Feedback Shift Register(LFSR)is a shift register with feedback path linearly related to the nodes using XOR gates. LFSRs are more popular because of their compact and simple design. Cellular Automata(CAs) are more complex to design but provide patterns with higher randomness. Cellular Automata(CAs) are mathematical models for complex systems containing large numbers of simple identical components with local interconnections that act together to produce complicated patterns of behavior. They consist of an n-dimensional lattice of cells, each with a finite set of possible values. A CA evolves in discrete time steps, and the value of a particular cell (local state) at any given clock cycle depends on the cell neighborhood values on the previous clock cycle, according to a specific rule(local rule).Both LFSR and CASR has advantages so in this paper Combination of LFSR and CASR is implemented to generate complex random number generator.

II. Linear Feedback Shift Register

An LFSR is a finite state machine that goes through n cycles before repeating the sequence. It is structurally a shift register with specific tap taken out and XOR-ed with nth flip-flop and fed back to itself. It can be represented as binary polynomial P(x). In this paper 4-bit LFSR is used whose maximum feedback polynomial is represented as X^4 $+X^3$+1 which produces 2^4-1=15 Pseudo Noise(PN) sequence. It is confirmed from the simulation waveform that 4-bit LFSR produces 15 random sequences it is shown in figure 3.The 4-bit LFSR architecture is shown in figure 1.

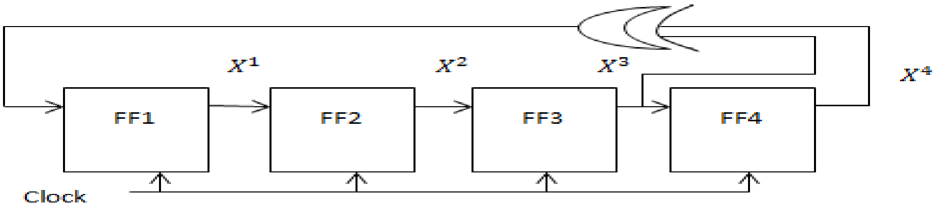


Figure 1.LFSR architecture

1.1 LFSR Simulation Results

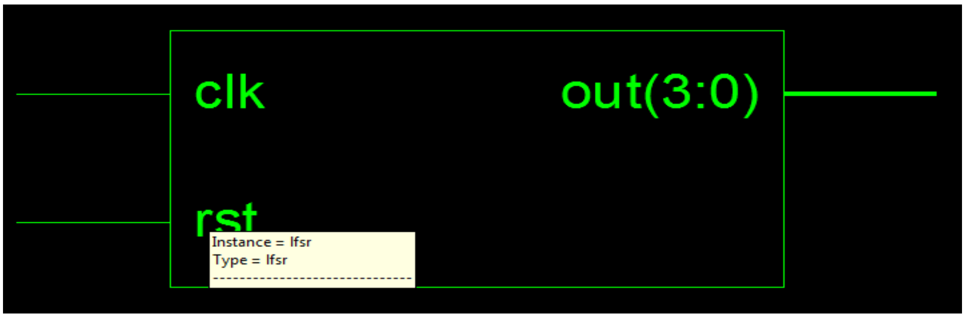


Figure 2.LFSR RTL Schematic

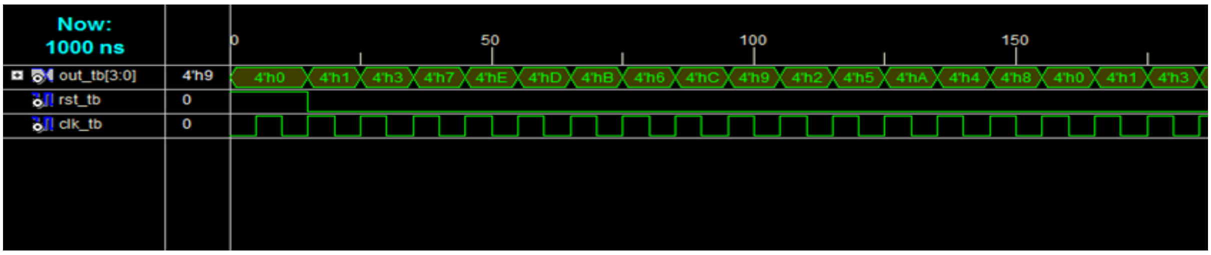


Figure 3.LFSR Simulation Result

III. Linear Congruential Generator

A Linear Congruential Generator (LCG) represents one of the oldest and best-known pseudo number algorithms. The theory behind them is easy to understand, and they are easily implemented and fast. However, their statistical properties are much worse than more recent generators, including generators with similar simplicity and speed.

The generator is defined by the recurrence relation:

X_{n+1}== (aX_n+c) mod m

Where,X_n is the sequence of pseudorandom values, m is the “modulus”, a is the “multiplier”, c is the “increment”,X_0 is the “seed” or “start value”.The period of a general LCG is at most m, and for some choices of a much less than that. Provided that c is nonzero, the LCG will have a full period for all seed values if and only if firstly c and m are relatively prime, secondly a-1 is divisible by all prime factors of m, and thirdly a-1 is a multiple of 4 if m is a multiple of 4.While LCGs are capable of producing decent pseudorandom numbers, this is extremely sensitive to the choice of the coefficients c, m, and a.

1.2 LCG Simulation Results

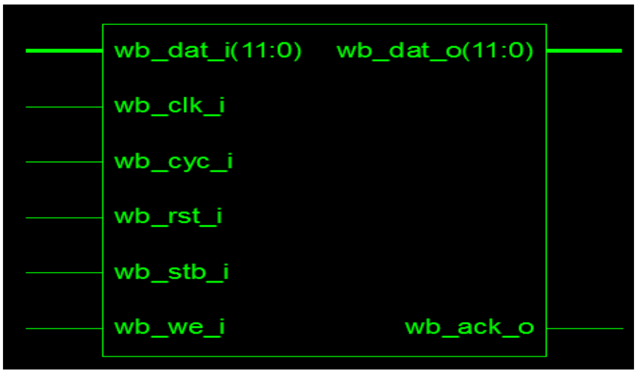


Figure 4.LCG RTL Schematic

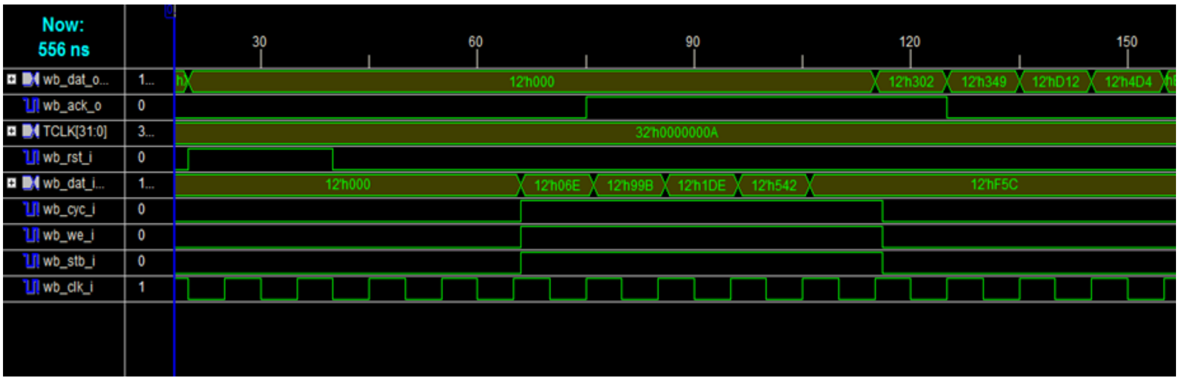
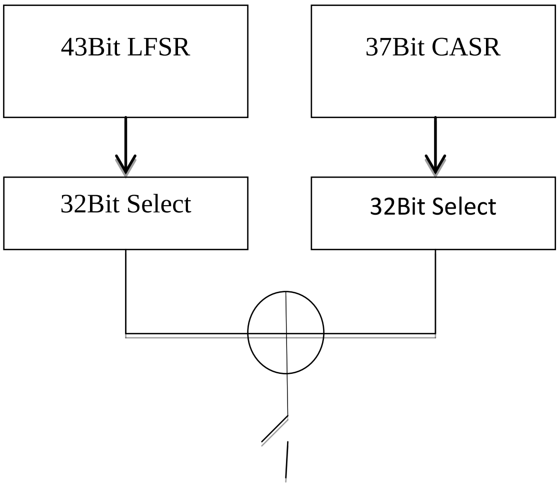


Figure 5.LCG Simulation Result

IV. Combination Of Lfsr And Casr

Cellular Automata consists of an n-dimensional lattice of cells, each with a finite set of possible values. LFSR is a shift register with feedback path linearly related to the nodes using XOR gates. LFSR is simple and have compact design whereas Cellular Automata is complex in design but provide patterns with higher randomness. To generate complex random number generator combination of LFSR and CASR is proposed in this paper. The LFSR has 43Bits and a characteristic polynomial of $X^{43} + X^{41} + X^{20} + X + 1$.This is primitive polynomial and gives a cycle length of 2^{43}-1.CASR has 37Bits with maximum length of 2^{37}-1.Combination is formed by permuting and XORing 32Bits of LFSR and CASR. The combination has a Cycle length of $2^{80} - 2^{43} - 2^{37} + 1$ as shown in figure 6.



Output

Figure 6. Combination of LFSR and CASR block diagram

3.1 Combination of LFSR and CASR Simulation Results

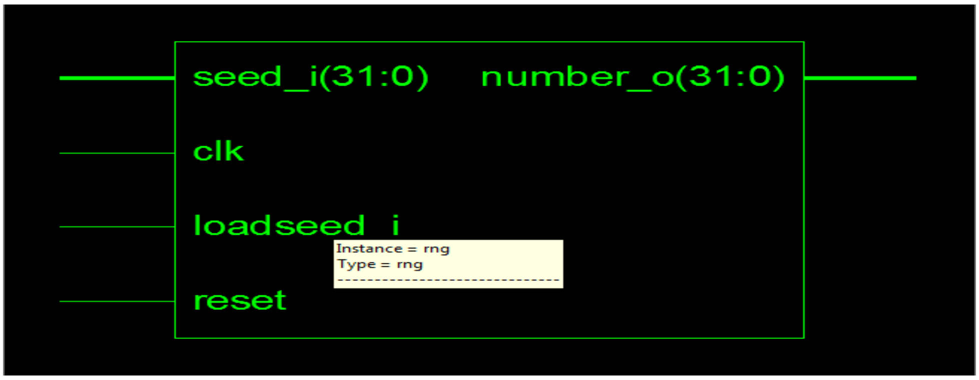


Figure 7. Combination of LFSR and CASR RTL Schematic

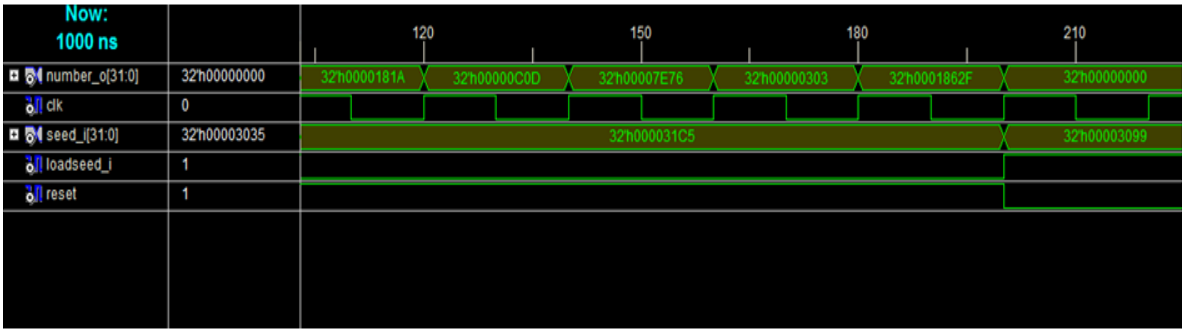


Figure 8. Combination of LFSR and CASR Simulation Result

V. Design Utilization Result Of Three Random Number Generators

Table 1: Design Results

LFSR

Logic Utilization	Used	Available	Utilization
Number of slice Registers	**4**	**93120**	**0%**
Number of slice LUTS	1	46560	0%
Number of fully used LUT-FF pairs	0	5	0%
Number of bonded IOBS	6	240	2%

LCG

Logic Utilization	Used	Available	Utilization
Number of slice Registers	**67**	**93120**	**0%**
Number of slice LUTS	97	46560	0%
Number of fully used LUT-FF pairs	52	112	46%
Number of bonded IOBS	30	240	12%

Combination of LFSR and CASR

Logic Utilization	Used	Available	Utilization
Number of slice Registers	**112**	**93120**	**0%**
Number of slice LUTS	113	46560	0%
Number of fully used LUT-FF pairs	112	113	99%
Number of bonded IOBS	67	240	27%

Table 2: Power results

On Chip Power comparison of LFSR, LCG and Combination of LFSR and CASR

Random number generators	Power(Watt)
LFSR	1.297
LCG	1.302
Combination of LFSR and CASR	1.327

VI. Conclusion

Several different ways have been already examined to increase randomness of random number generator. For a single bit random number generator, LFSR is most effective method. LFSR is very low memory cost consuming. LCG require minimal memory to retain state. LCGs should not be used for applications where high quality random is critical.LCG is slow and expensive in terms of silicon area since multiplier, division and addition is used.LFSR and LCG are linear structure and output of these RNGs are predictable. To avoid predictable nature,Combination of LFSR and CASR is proposed in this paper. It gives longer cycle length and provide patterns with higher randomness and also it is highly secure. RTL schematic of these RNGs using Verilog language are implemented. Comparison of design utilization of these RNGs are done in this paper by using Xilinx ISE 10.1 simulator.

References

[1] M. Luby, Pseudorandomness and Cryptographic Applications, Princeton University Press, 1996.

[2] Random Number Generator (2011). Wikipedia website [Online]. Available:http://en.wikipedia.org/wiki/Hardware_random_number_generator.

[3] Jiang Hao, Li Zheying, "On the Production of Pseudo-random Numbers in Cryptography" in Journal Of Changzhou Teachers College of Technology, Vo1.7 , No. 4, Dec. 2001.

[4] D. E. Knuth, "The Art of Computer Programming", Vol. 2: Seminumerical Algorithms. Reading, MA: Addison-Wesley, 1969.

[5] F. James, “A Review of Pseudo-random Number Generators,” Computer Physics Communications 60, 1990.
[6] P. L'Ecuyer, “Random Numbers for Simulation,” Comm. ACM, 33:10, 1990.
[7] Pseudo-Random-Generator. Wikipedia website [Online] Available: http://en.wikipedia.org/wiki/Pseudorandom_number_generator
[8] Katti, R.S. Srinivasan, S.K., “Efficient hardware implementation of a new pseudo-random bit sequence generator” IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009.
[9] C. Li and B. Sun, “Using linear congruential generators for cryptographic purposes”, In Proceedings of the ISCA 20th International Conference on Computers and Their Applications, pp. 13-18, March 2005.
[10] L'Ecuyer, Pierre, “Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure,” Mathematics of Computation, Vol. 68, No. 225, 1999, Pages 249-260.

IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) is UGC approved Journal with Sl. No. 5016, Journal no. 49082.

Shruthi K. "Comparative Analysis of High Efficient Random Number Generator." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 12.4 (2017): 87-91.