Pseudo-Random Number Generation by Fibonacci and Galois LFSR Implemented on FPGA

Pritish A.Deshmukh
M-Tech Student
B.D.C.O.E.Sewagram
Wardha-India

Y.A. Sadawarte
Assistant Professor
B.D.C.O.E. Sewagram
Wardha-India

ABSTRACT
Random Number Generator is an electronic circuit or it can be software or can be optimized architecture. In many practical applications such as cryptography, model simulation, sampling, games of chance, numerical analysis, there is a need of the generation of series of random number. This is achieved for ex. by means of tables, specific algorithms or electronic circuits. This Random number can be generated by either specific software or a FPGA based architectures. Field Programmable Gate Array (FPGA) optimized random number generator (RNG) are more resource efficient than software optimized RNG because they can take the advantage of bitwise operations and FPGA specific features. Hence for generation of random number, FPGA architecture is generally used. By using different FPGA platform, random number can be generated. There are several algorithms by which the random number has been generated. Each algorithm had used a different FPGA platform for generation of random number. In this paper generation of 8 bit random number by means of two method i.e Fibonacci series method and Galois liner feedback shift register method. Moreover for analysis Altera platform is used i.e. for simulation Modelsim software and for synthesis Quartus II software is used. Hence by using these two algorithms, random numbers have been generated and each algorithm has shown different performance parameters i.e. area, speed and power.

Keywords
Random number generator RNG, Field programmable gate array FPGA, Linear Feedback Shift Register LFSR.

1. INTRODUCTION
Random Number Generator is an electronic circuit or it can be software or can be optimized architecture. In many practical applications such as cryptography, model simulation, sampling, games of chance, numerical analysis, there is a need of the generation of series of random number. This is achieved for ex. by means of tables, specific Algorithm or electronic circuits. Moreover there are again some FPGA architecture by which we can generate the series of random number. Each algorithm Use a different FPGA platform and the random number can be generated. FPGA optimized RNG are more resource efficient than Software optimized RNG because they can take Advantages of bitwise operation and FPGA specific features These FPGA architectures shows huge performance in terms of area and speed. In this paper the 8 bit random number is generated by two methods i.e. Fibonacci series method and Galois liner feedback shift register method.

1. INTRODUCTION ABOUT LINEAR FEEDBACK SHIFT REGISTER

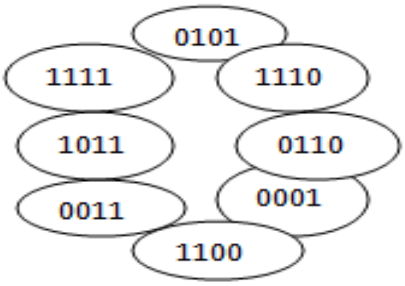

Figure 1. Block Diagram of Basic Linear Feedback Shift Register

In computing, a linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters and whitening sequences.

2. PSEUDORANDOM NUMBER GENERATOR BASED ON FIBONACCI SERIES METHOD

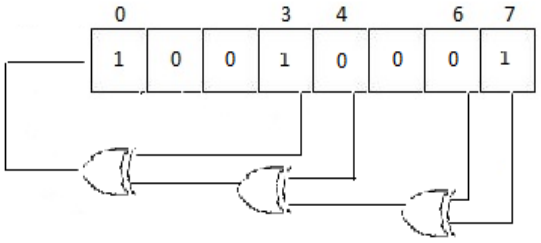

Figure 2. Block Diagram of Fibonacci Series Method

The Bit positions that affect the next state are called the taps. In the diagram the taps are [7, 6, 4, 3] .The rightmost bit of

the LFSR is called the output bit. The taps are XOR'd sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream.

A maximum length LFSR produces an m-sequence i.e. it cycles through all possible 2^n - 1 state. The sequence of numbers generated by an LFSR can be considered a binary numeral system just as valid as Gray code or the natural binary code. The arrangement of taps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod.

This means that the coefficients of the polynomial must be 1's or 0's. This is called the feedback polynomial or reciprocal characteristic polynomial. For example, if the taps are at the 7th, 6th, 4th and 3th bits (as shown), the feedback polynomial is $X^7+ x^6 + x^4 + x^3 +1$. The 'one' in the polynomial does not correspond to a tap — it corresponds to the input to the first bit (i.e. x0, which is equivalent to 1).

The powers of the terms represent the tapped bits, counting from the left. The first and last bits are always connected as an input and output tap respectively.

The LFSR is maximal-length if and only if the corresponding feedback polynomial is primitive. This means that the following conditions are necessary (but not sufficient):

The number of taps should be even.

The set of taps — taken all together, not pair wise (i.e. as pairs of elements) — must be relatively prime. In other words, there must be no divisor other than 1 common to all taps.

3. PSEUDORANDOM NUMBER GENERATOR BASED ON GALOIS LFSR METHOD

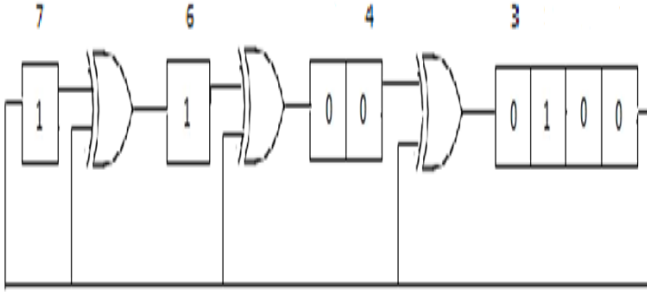

Figure 3 Block Diagram of Galois LFSR Method

In the Galois configuration, when the system is clocked, bits that are not taps are shifted one position to the right unchanged. The taps, on the other hand, are XOR'd with the output bit before they are stored in the next position. The new output bit is the next input bit

The effect of this is that when the output bit is zero all the bits in the register shift to the right unchanged, and the input bit becomes zero.

When the output bit is one, the bits in the tap positions all flip (if they are 0, they become 1, and if they are 1, they become 0), and then the entire register is shifted to the right and the input bit becomes 1.

Galois LFSRs do not concatenate every tap to produce the new input (the XOR'ing is done within the LFSR and no XOR gates are run in serial, therefore the propagation times are reduced to that of one XOR rather than a whole chain), thus it is possible for each tap to be computed in parallel, increasing the speed of execution.

In a software implementation of an LFSR, the Galois form is more efficient as the XOR operations can be implemented a word at a time: only the output bit must be examined individu

4. SIMULATION AND SYNTHESIS RESULTS

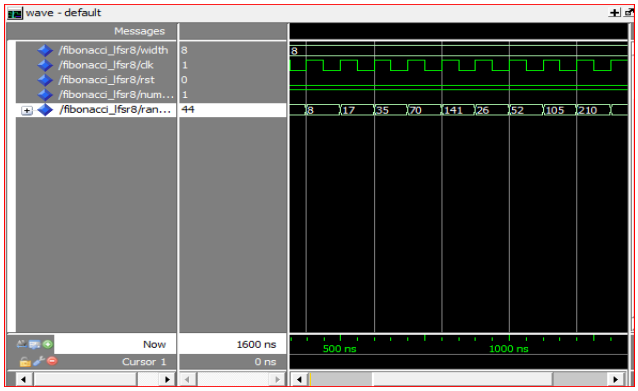

Figure 4. Simulation Result of Fibonacci Series Method

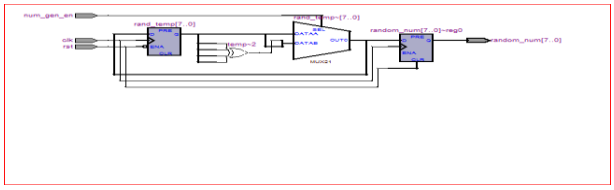

Figure 5. RTL View of Fibonacci Series Method

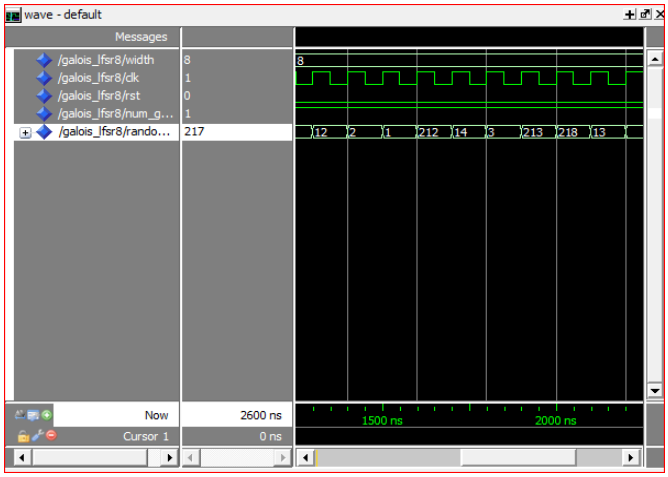

Figure 6. Simulation Result of Galois LFSR Method

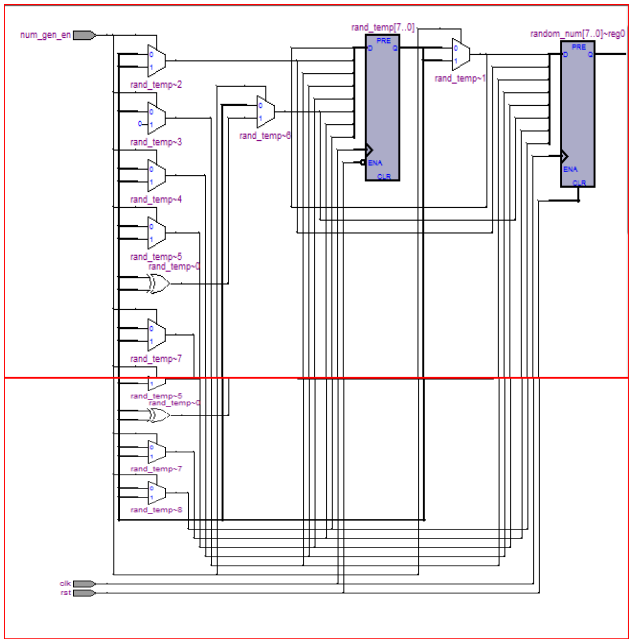

Figure 7. RTL View of Galois LFSR Method

For simulation and synthesis of the algorithm Altera platform is used, Modelsim software is used for simulation of the two methods and for synthesis Quartus II software is used. In that Quartus II 9.1 build 350 03/24 /2010 SP 2 SJ web edition version is used. Moreover the family is cyclone II and the device is EP2C20F484C7 is used.

Table 1 Comparison Table

Method	Fibonacci	Galois LFSR
Total logic elements	08/18752 (< 1 %)	07/18752 (<1 %)
Power Dissipation	20.61 Mw	21.57 Mw
Time delay	9.39 ns	8.17 s

5. CONCLUSION

In this paper, we have successfully demonstrated the design of high speed, low power random number generator based on Fibonacci and Galois LFSR. Therefore this design is well suited for high performance and low power requirement.

6. REFERENCES

[1] Ravi saini, Sanjay Singh, Anil Saini, AS Mandal, Chandra Shekhar CSIR- central electronics engineering research institutes (CSIR-CEERI) Pilani-Rajasthan, India 2013 on Design of a Fast and Efficient Hardware Implementation of a Random Number Generator in FPGA

[2] carols Gayoso, C.gonzalez in 2013 on Pseudorandom Number Generator Based on the Residue Number System and Its FPGA Implementation in international conference on advance electronic system (ICAES)

[3] Yuan li, Paul Chow, Senior member IEEE, Jiang, Minxuan zhang, and shaojun wei in 2013 on Software / Hardware Parallel Long Period Random Number Generation Framework Based On The Well Method in IEEE Transactions

[4] David b. Thomas, member of IEEE and Wayne luk, fellow in April 2013 on A LUT-SR Family of Uniform Random Number Generators for FPGA Architecture IEEE transactions on very large scale integration system, Vol 21 no 4

[5] Jonathan M. Comer, Juan C. Cerda, Chris D. Martinez, and David H. K. Hoe in 2012 on Random Number Generators Using Cellular Automata Implemented on FPGA

[6] Ray C. C. Cheung, Student Member, IEEE, Dong-U Lee, Member, IEEE, Wayne Luk, Senior Member, IEEE ,in 2007 on Hardware Generation of Arbitrary Random Number Distributions from Uniform Distributions Via the Inversion Method

[7] jiang hanging, shaojun wei international conference on computer and information technology in 2012 on An efficient hardware random number generator based on MT method

[8] N.szaboo and R Tanaka in 1967 on Residue arithmetic and its application to computer technology M.soderstand, w Jenkins, jullien and F.Taylor in 1986 , residue number system arithmetic modern application in digital signal processing

[9] Savir,, a new empirical test for quality of random integer generators

[10] Residue arithmetic a tutorial with examples. Computer magazines IEEE vol 1

[11] C.M. Gonzalez, H.A.larrondo Xi workshop Implementations of de sistemas caotiocs en dispostivos logicos programmable

[12] Intensive stastical complexity measures of pseudorandom bit" by C.M. Gonzalez, H.A.larrondo

[13] A. Ross, H.larrondo, M mertin Generalized stastical complexity measures a new tool for dynamic systems [14]Altera corporation www.altera.com FLEX 10K embedded programmable logic device family data sheet 2001

[14] F. Panneton, M.Mastumoto Improved long period generators based on the linear recurrences modulo 2 overview and comparison.

[15] Yuan li, Paul Chow, Senior member IEEE, Jiang, Minxuan zhang, and shaojun wei in 2013 on Hardware Parallel Long Period Random Number Generation Framework Based On The Well Method" IEEE Transactions On Very High Speed Large Scale Integration.