RES MILITARIS Social Science Journal

PSEUDO RANDOM NUMBER GENERATOR IN FPGA AND CMOS VLSI

[1]W.Roshini,[2]C.Chandra Kiran,[3]N. Ramya,[4]B.Vinay,[5]H.Somashekar
[1,2,3,4]IV Year Student,[5]Assistant Professor
Department Of ECE
Visvesvaraya College Of Engineering & Technology,Ibrahimpatnam,Telangana

ABSTRACT
Pseudo Random number Generator (PRNG) is used in various cryptographic applications such as Bank Security, Generation of Keys which are used for Encrypting or Decrypting Messages, Networking etc. The Random number generator discussed in this paper uses the concept of Shift registers and is designed using Maximum length feedback polynomial, which is more advantageous when compared to other Random number generators or Counters that are used in Cryptography. In this paper the design and implementation of PRNG using Field programmable gate array (FPGA) is explained and it is compared with other counters to observe its performance in various aspects. PRNG is also implemented using different topologies in CMOS to reduce the Power Consumption

I. INTRODUCTION TO RANDOM NUMBER GENERATORS

A random number generator is a hardware device or software algorithm that generates a number that is taken from a limited or unlimited distribution and outputs it. The two main types of random number generators are pseudo random number generators and true random number generators.There are generally two types of random number generators: pseudorandom number generators and true random number generators.

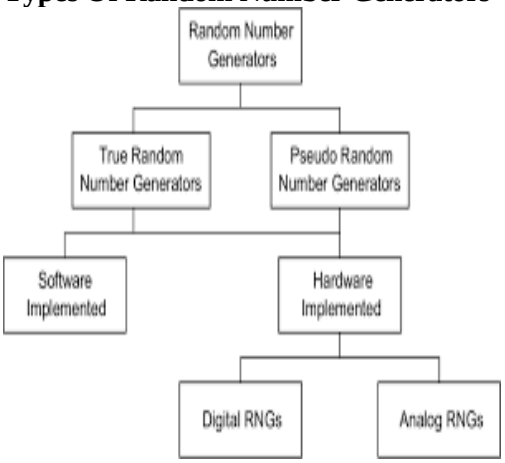
Pseudo Random Number Generators

Random number generators are typically software, pseudo random number generators. Their outputs are not truly random numbers. Instead, they rely on algorithms to mimic the selection of a value to approximate true randomness. Pseudo random number generators work with the user setting the distribution, or scope from which the random number is selected (e.g., lowest to highest), and the number is instantly presented.The outputted values from a pseudo random number are adequate for use in most applications but they should not always be relied on for secure cryptographic implementations. For such uses, a cryptographically secure pseudo random number generator.

True Random Number Generators

A true random number generator — a hardware random number generator (HRNG) or true random number generator (TRNG) — is cryptographically secure and takes into account physical attributes such as atmospheric or thermal conditions. Such tools may also take into account measurement biases. They may also utilize physical coin flipping and dice rolling processes. A TRNG or HRNG is useful for creating seed tokens.

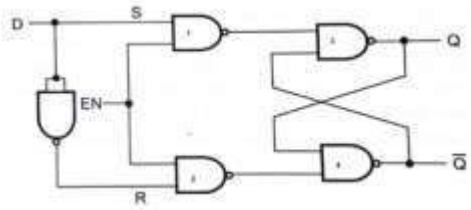
Types Of Random Number Generators



D Flip Flop

A D (or Delay) Flip Flop is a digital electronic circuit used to delay the change of state of its output signal (Q) until the next rising edge of a clock timing input signal occurs.

D Flip flops or data flip flops or delay flip flops can be designed using SR FLIPFLOP, by connecting a not gate in between S and R inputs and tying them together. D flip flops can be



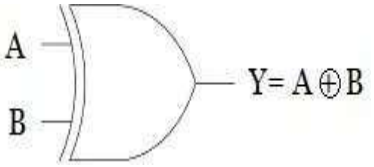
We know that the SR flip-flop requires two inputs, i.e., one to "SET" the output and another to "RESET" the output. By using an inverter, we can set and reset the outputs with only one input as now the two input signals complement each other. In SR flip flop, when both the inputs are 0, that state is no longer possible. It is an ambiguity that is removed by the complement in D-flip flop.
In D flip flop, the single input "D" is referred to as the "Data" input. When the data input is set to 1, the flip flop would be set, and when it is set to 0, the flip flop would change and become reset. However, this would be pointless since the output of the flip flop would always change on every pulse applied to this data input.
The "CLOCK" or "ENABLE" input is used to avoid this for isolating the data input from the flip flop's latching circuitry. When the clock input is set to true, the D input condition is only copied to the output Q. This forms the basis of another sequential device referred to as **D Flip Flop**.
When the clock input is set to 1, the "set" and "reset" inputs of the flip-flop are both set to 1. So it will not change the state and store the data present on its output before the clock transition occurred. In simple words, the output is "latched" at either 0 or 1.

- **Truth Table for the D-type Flip Flop**

Clock	D	Q	Q'	Description
↓ x ⇕	X	Q	Q'	Memory no change
↑ x 1	0	0	1	Reset Q = 0
↑ x 1	1	1	0	Set Q = 1

XOR Gate:
XOR and XNOR gates are special-purpose gates. They can be used for applications such as half adder, full adder, and subtractor.
These gates are also called derived gates.
The XOR Gate is abbreviated as EX-OR gate or sometimes as Exclusive-OR gate.
An XOR gate can have two or more two inputs terminals and one output terminal.

The XOR gate symbol is shown in the figure below.



The **XOR Gate Truth Table** is as shown in the figure below which shows that, when both the inputs are identical (A=B), the output is LOW (0) i.e. Y=0 for A=B=0 or A=B=1, and the output is HIGH (1) when A≠B.

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

The Boolean expression of a xor gate is:
$Y=A\bar{B}+A\bar{B}$
Some of the applications of XOR gates are as follows:

- As a magnitude Comparator.
- In the binary-to-gray code converter.
- used in adder and subtractor circuits.
- In the parity generator.
- as a modulo-2 adder.

II. LITERATURE SURVEY:

"Multiple output Low Power Linear Feedback Shift Register Design,"
In this paper, we present a new low-power architecture for linear feedback shift registers (LFSRs) that produces the output of several clock cycles of a serial LFSR at once while reducing the activity factors of the flip-flop outputs. The frequency of operation can thus be reduced by a factor equal to the number of outputs produced at a time. A reduction in the frequency of the LFSR allows for a reduction in the power-supply voltage. Thus, dynamic power dissipation is reduced by up to 93% due to decreases in power-supply voltage, frequency, and the activity factor.

Furthermore, the hardware needed for our implementation is far less than previous low-power implementations of both single and multiple-output LFSRs. Our method is also good for built-in self-test (BIST) applications because for most degrees of N it results in all 2/sup N/-1 distinct patterns.

"Design of Multi Bit LFSR PNRG and Performance comparison on FPGA using VHDL"

There are many applications of linear feedback shift registers, and it is commonly used in mobile communications where pseudo random sequence is required. These are the basic blocks of many circuits like PN sequence generator, gold code producer which is used in spread spectrum code division multiple access techniques. Linear feedback shift registers are extensively used for binary counters to generate random number sequences. Maximum time the generated sequence is pseudorandom in nature. These patterns may repeat over period, as longer the shift registers. This repetition is depends on the number of taps present in the registers. For large pattern generation, the size of hardware may be increased. Conservatively, for the older architectures of FPGA, flip flops were used. LFSR sequences are generated through 2N – 1 state, where N is the number of flip-flops/taps in the LFSR. After every edge of clock, the data of flip flop is shifted right. The feedback path is provided from previous register to the left most register through an XOR or XNOR gate. Value of 0's is illegal for XOR feedback path similarly value of 1's is illegal for XNOR feedback path. These illegal states may cause the shift register to present in its present state [13]. A 4-bit LFSR sequences generated through (24 - 1) is having 15 states (the state 1111 is in the illegal state) from the feedback taps 4 and 3. At the same time, a 4 bit binary counter may generate the sequence by 24 i.e. 16 states without any illegal stages. Still linear feedback shift register are faster than normal counter because they don't have carry signal. Linear feedback shift register are the substitute of normal binary counters in perilous applications where the counted sequence is not that much important. Linear feedback shift register are also used as pseudo random sequence generators. These are the basic blocks of many circuits like PN sequence generator, gold code producer which is used in spread spectrum code division multiple access techniques. The tap sequence is responsible to affect the bits positions of next bits. The n bit LFSR whose maximum feedback polynomial is represented as follows:

III. EXISTING ARCHITECTURE:

LSFR (linear shift feedback register)

A linear feedback shift register (LFSR) is a shift register whose input bit is the output of a linear function of two or more of its previous states

3.2 *4*-bit LFSR

•Circuit counts through 24-1 different non-zero-bit patterns.

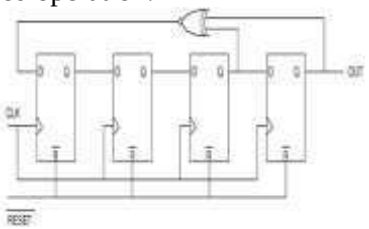
• Leftmost bit decides whether the "10011" XOR pattern is used to compute the next value or if the register just shifts left. • Can build a similar circuit with any number of FFs, may need more XOR gates.

• These are n-bit counters exhibiting pseudo-random behavior.

• Built from simple shift-registers with a small number of XOR gates

• Used for: – pseudo-random number generation – counters – error checking and correction

•Advantages: – very little hardware – high speed operation.



4-bit LFSR • Circuit counts through 2 4 -1 different non-zero-bit patterns. •Leftmost bit decides whether the "10011" XOR pattern is used to compute the next value or if the register just shifts left.

• Can build a similar circuit with any number of FFs, may need more XOR gates.

•In general, with n flip-flops, 2 n -1 different non-zero-bit patterns. 0 0 0 1 0 XOR 0 •0 0 0 0 0 0 1 0 0 XOR 0 0 0 0 0 0 0 1 0 0 0 XOR 0 0 0 0 0 0 1 0 0 0 0 XOR 1 0 0 1 1 0 0 0 1 1 0 XOR 0 0 0 0 0 0 0 1 1 0 0 XOR 0

0 0 0 0 0 1 1 0 0 0 XOR1 0 0 1 1 0 1 0 1 1 Q4 Q3 Q2 Q1 0001 0

3.3 The theory behind LFSRs:

• LFSR circuits performs multiplication on a field.

• A field is defined as a set with the following: – two operations defined on it:

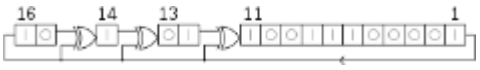
• "addition" and "multiplication" – closed under these operations – associative and distributive laws hold – additive and multiplicative identity elements – additive inverse for every element – multiplicative inverse for every non-zero element

• Example fields: – set of rational numbers – set of real numbers – set of integers is not a field

• Finite fields are called Galois fields. • Example: – Binary numbers 0,1 with XOR as "addition" and as "multiplication". – Called GF (2)

An LFSR is essentially a shift register where a few of the bits are set to result of XOR other bits in the register. This results in the shift register producing a pseudo-random series of numbers. With careful construction, the pattern can be made to cycle through all possible combinations of bits (except for all 0s) before repeating.

Your 4-bit LFSR should use the design below. When reset, the value should be 0001.



XOR operations can be implemented a word at a time: only the output bit must A 16-bit Galois LFSR. The register numbers above correspond to the same primitive polynomial as the Fibonacci example but are counted in reverse to the shifting direction. This register also cycles through the.

XOR operations can be implemented a word at a time: only the output bit must A 16-bit Galois LFSR. The register numbers above correspond to the same primitive polynomial as the Fibonacci example but are counted in reverse to the shifting direction. This register also cycles through the

IV. PROPOSED ARCHITECTURE:

PSUEDO RANDOM NUMBER GENERATOR

In Shift Register input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value given to the Shift Register is called the seed [1]. The feedback function should be selected in such a way so that it produces a sequence of bits which has a very long cycle and random in nature. The repeating sequence of a Shift Register allows it to be used as a Counter, or as a Random number generator however it is necessary to ensure that the Shift Register never enters an all-zeros state. PRNG designed using Shift Registers have simpler Feedback logic than natural gray counters or Binary code counters, and operates at higher clock rates [2]. Binary counters are designed using Flip-Flops, half adders, and a high-speed carry chain. The delay associated with such counters depends on the number of bits in the adder/carry chain circuit. In contrast, PRNG designed using Shift Register use only XOR gates and Flip-Flops. The delay associated is independent of the number of bits in the counter. Binary, Gray Counters suffer from the problems of glitches, speed, Power Consumption and delay. They produce not only glitches, which increase power consumption but also increases the complexity of design

4.2 MAXIMUM LENGTH FEEDBACK POLYNOMIAL

Maximum-length Feedback Shift Register produces a maximum sequence i.e., it cycles through all the possible 2n - 1 states within the Shift Register [4]. The only signal necessary to generate the Random patterns is the Clock. Shift Register outputs are traditionally labelled from 1 through n, with 1 being the first stage of the shift register,

and n being the last stage [4]. This is different from the conventional 0 to (n-1) notation for binary counters. A. Rules for Selecting Maximum Length Feedback Polynomial Shift Registers produce the Maximum Length sequence, when the characteristic polynomial used in the design is of Maximum Length [3]. The choice of Shift register length, gate type, Maximum length logic, and tap positions allows the user to control the implementation and feedback of the Shift Register, which, in turn, controls the sequence. The following things have to be noted while selecting the Maximum Length Feedback Polynomial The initial value of the Shift Register is called the¬ seed. The seed value can be anything except all 0s i.e., the Pseudo Random sequence must start in a non-zero sequence. The 'One' in the Maximum length Feedback¬ sequence corresponds to the principal input of the shift register. The Shift Register will only be Maximum length if¬ Tap values should the numbers of taps are even. be relatively prime. The first and last taps should always be connected as input and output taps respectively. Mirror Sequence exists for the given tap sequence¬ and can be more than one tap sequence for a particular.

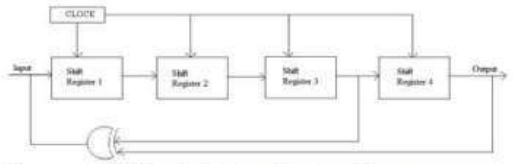


Figure 1: 4 Bit Pseudo Random Number Generator

The above Figure is the basis for the design of PRNG in both FPGA and in CMOS VLSI. The design is carried out in two phases. In the first phase the Circuit is designed and implemented in FPGA. The target device used in the design is Xilinx Spartan XC3S 500e. Simulation and Synthesis is performed using Xilinx and the results are compared with other counters. Various parameters like Speed, number of Flip-Flops required, LUTS required, Delay etc. are compared in this phase and Verilog HDL is preferred for programming because it is less complex and widely used. In the second phase the paper mainly focuses on the transistor level design of the PRNG. This is performed using Mentor Graphics HEP-1 tool.

DESIGN ASPECTS OF PRNG

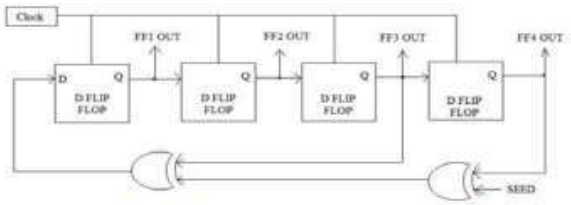


Figure 2: 4 Bit Pseudo Random Number Generator (Flip Flop Model)

The PRNG designed in the Figure 1 is a combination of Shift Registers and XOR Gates. Shift Registers are nothing but Flip Flops. By changing the Flip Flop Designs, the power consumption of the PRNG can also be reduced to a considerable extent finally producing a low power consuming VLSI Chip [3]. There are many Topologies for the design of Flip-Flops. Some of the prominent Flip-Flop Topologies have been discussed in this paper.

V. ADVANTAGES & APPLICATIONS:

ADVANTAGES: -

- **Efficient**: PRNG can produce many numbers in a short time and is advantageous for applications that need many numbers.
- **Deterministic**: A given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known.
- When we talk about practical usage, the results obtained from most of the common types of PRNGs are essentially inconsistent. This is owing to the following reasons:
- Some seed states have very short periods. (Such seeds are often termed 'weak').
- Non-uniformity in distribution in most of the generated numbers
- Correlation of successive values;
- Output sequence lacking proper dimensional distribution;

- Inconsistency in distance in certain distributions not adhering to the random sequence distribution. Sometimes the defects obtained from PRNGs may go unnoticed while sometimes they seem to be quite obvious. A classic example of such kind is the algorithm RANDU that had serious defects even though it remained unnoticeable for several years.

APPLICATIONS: -

Simulation: Replicating natural phenomenon using random numbers. Some beautiful examples of simulation come under the fields of Physics and Chemistry.

Sampling: Sometimes it is very difficult to consider the numerous cases as a whole. In such cases, choosing a specific sample from the huge lot makes it easier.

Computer programming: A programs effectiveness can be assessed by feed in it several random data.

Cryptography: This domain finds a plethora of applications as in large random integers, session keys, etc.

Decision making: Over the ages random numbers have found vital applications in decision making scenarios some of the classical examples being rolling a die or tossing a coin. 3

VI. RESULT:

VII. CONCLUSION:

Shift Register based Pseudo Random Number Generator has many advantages over other counters in terms of Speed, Area and Hardware. It is preferable when good logic density is to be maintained during fabrication process, while obtaining power optimization and reducing the propagation delay & glitches of the circuit. The Comparison results of different Counters with Shift Register based PRNG is provided in this paper and it is implemented using different Flip-Flop Topologies. The power consumption for different topologies is verified using Mentor Graphics tool. The power consumption can further be reduced by using other available topologies there by making it more efficient for the chip design.

FUTURE SCOPE

A pseudo-random number generator (PRNG) is a program written for, and used in, probability and statistics applications when large quantities of random digits are needed. Most of these programs produce endless strings of single-digit numbers, usually in base 10, known as the decimal system.

We can also extend this design **LSFR** in 8-bit and as well as 16-bit also.

REFERENCES

[1]. Rajendra S. Katti, Xiaoyu Ruan and Hareesh Khattri, “Multiple output Low Power Linear Feedback Shift Register Design," IEEE Transactions on Circuits and Systems-I, vol. 53, No.7 July 2006.

[2]. Shiv Dutta Mishra, Prof. Anurag Shrivastav “Design and Analysis of FPGA based cryptographic N-bit parallel LFSR”, International Journal of Latest Trends in Engineering & Technology (IJLTET), NOV 2013, Vol. 3, Issue 2, ISSN. 2278-621X.

[3] Goresky, M. and Klapper, A.M. Fibonacci and Galois representations of feedback-with-carry shift registers, IEEE Transactions on Information Theory, Nov 2002, Volume: 48, On page(s): 2826 – 2836.

[4] Panda Amit K, Rajput P, Shukla B, “Design of Multi Bit LFSR PNRG and Performance comparison on FPGA using VHDL”, International Journal of Advances in Engineering & Technology (IJAET), Mar 2012, Vol. 3, Issue 1, pp. 566-571.

[5]. Doshi N. A., Dhobale S. B., and Kakade S. R., “LFSR Counter Implementation in CMOS VLSI”, World academy of Science

and Technology, 48 2008. Chandra Sekhar.K et al, / (IJCSIT) International Jou