International Journal of Computer Applications (0975 – 8887)
Volume 138 – No.3, March 2016

Linear Feed Back Shift Register Base Multiple Long Period Random Binary Sequences Generator

Neha Agrawal
Department of Electronics & Communication
T.I.E.I.T
Bhopal, M.P., India

Neelesh Gupta
Department of Electronics & Communication
T.I.E.I.T
Bhopal, M.P., India

Neetu Sharma
Department of Electronics & Communication
T.I.E.I.T
Bhopal, M.P., India

ABSTRACT

In electronics communication systems random number generation is use for security purpose. The numbers which are in random but in predefined sequence pattern is called as pseudo random while the numbers which are unpredictable and in undefined sequence pattern is called as true random numbers. These random numbers are also use for bit error rate testinig (BERT). When multiple bits are required then linear feedback shift registers are the best source of random number generator. The increase in length of random number sequence consumes more area. Here a increase lenth of sequence and multiple bits random number generator is design using linear feedback shift registers and multiple port SRAM memory. The SRAM base random number generator is area efficient using VHDL. The improved computational time and throughput is computed using VHDL implementation.

Keywords

RNG, BERT, LFSR , SRAM

1. INTRODUCTION

For the bit error rate testing in digital communication pseudo random numbers binary sequence is mostly used. The pseudo word means that the random sequence repeat itself but it is not truly random. Pseudo random binary sequence can be created by shifting bits through a N number of series connected registers, in which outputs of each flip flops called as tap are feedback to the input of first flip flop through XOR logic gate to design the linear feedback shift register. The length of the sequence is increase which determined by the number of all possible states except all zeros through which LFSR state changes and the properties of the sequence is determined by which tap sets that are (mod2) added and feedback to the first register. This type of pseudo random binary sequence generator is called a linear feedback shift Register (LFSR). This work presents the VHDL implementation for SRAM base linear feedback shift register long-period pseudo random number generator. Random number generator is use for strengthening and securing data of electronics communication. LFSR is the most efficient method for one bit random number generator. When many bits are required, LFSR can be extended by design with extra circuitry.

2. PROPOSED SYSTEM

Pseudo random number generations are algorithms that generated a number sequence in random way by using an initial condition and seed. The initial condition will decide where the sequence will initialize. The seed decides the pattern which the sequence will follow.

In work publish by Gianluca Setti, and Riccardo Rovatti highlights the drawbacks of the classical solutions and general features of an RNG, showing how they can be overcome by the chaos-based approach. Work review pipeline ADCs, in particular the 11/2 bit per stage architecture which appears to be the best suited for the reuse of building blocks in chaos-based true random generators. Work shows that the design is tamper resistant in the sense that a power analysis cannot leak information regarding the generated bits [1]. David B. Thomas and Wayne Luk did there work on bitwise operations, lookup tables LUTs and Field-programmable gate array (FPGA) optimized random number generators (RNGs). Design bitwise XOR operations and the ability to turn lookup tables (LUTs) into shift registers of varying lengths using LUT, FIFO (First in first out) memory [2]. Walter Aloisi and Rosario Mita work on method to reduce the power consumption of the popular linear feedback shift register. The proposed scheme is based on the gated clock design approach and it can offer a significant power reduction, depending on technological characteristics of the employed gates. The efficiency of the gated-clock LFSR design depends on the technological parameters of the adopted gates [3].

3. SYSTEM DESIGN

Increase number of bits in generator by using a Linear feedback shift register and making measurements on VHDL simulator. Finite state machine is use to generate wide range of code with small amount of logic using VHDL. LFSR circuit required series connected block of D flip-flop, XOR gate and or gate. Seed are important and had been used as input pattern for LFSR counter. Based on hardware requirement, the most essential parts are D-flip flops. Different D flip-flop methodology will give difference power consumption and layout sizing. The maximum length sequence of an LFSR is the longest number of cycles in the LFSR until the generated pattern repeats itself. One major characteristic of the maximum length sequence is that every state will only be entered once between repetitions. Thus, for an N-bit LFSR with appropriate feedback taps selected, the sequence will take 2n-1 clock cycles to repeat. The missing cycle is due to the lockup state of the feedback pattern.

LFSRs sequence through 2N –1 states, where N is the number of registers in the LFSR. The contents of the registers are shifted right by one position at each clock cycle. The feedbacks from predefined registers or taps to the leftmost register are XORed together. LFSRs have several variables:

- The number of stages in the shift registers
- The number of taps in the feedback path
- The position of each tap in the shift registers stage
- The initial starting condition of the shift register, often referred to as the FILL state

4. PROBLEM STATEMENT

The problem that is to be worked was the large no. of components used is the previous system. In this proposed system, the no. of components viz. flip-flops, registers are reduced. Also, the computation time ,throughput & latency gets improved.

5. PROPOSED METHODOLOGY

A. VHDL Implementation of LFSR's:

The linear feedback shift register is design in two modules, one is a shift register and second is a feedback function. The shift register is started with n bits (initialized sequence), and every time a key stream bit is essential, every bits in the register are shifted 1 bit to the right. So the LSB (least significant bit) is the output bit. The left-most bit is calculated as the XOR of definite bits in the register. This structure can potentially produce a 2n-1 bit-long pseudo-random sequence before repeating.

The fig 1 shows the timing simulation of static read write acces memory base random number generation and its RTL view generated by VHDL.

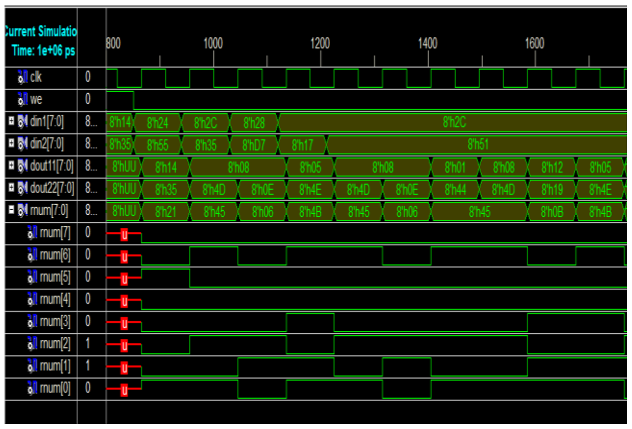


Fig.1 Timing simulation of propose SRAM base RNG random number generate operation.

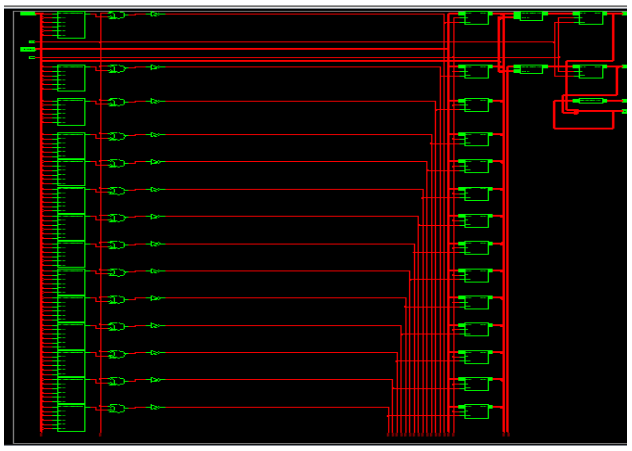


Fig.2 RTL view of propose SRAM base RNG random number generate operation

6. SIMULATION RESULTS

The proposed design is implemented using the VHDL hardware description language. The implementation supports a range of parameters to facilitate the experimental evaluation of design choices. A functional, parameterizable implementation is then produced, based on the design specification. The analysis of design decisions suggests that some design options provide no advantages over their alternatives. Such options are not supported in the implementation. The design is implemented using VHDL, a standard hardware description language. The implementation serves two main purposes. First, it verifies the correctness of the design. Second, it allows for verifying the analysis experimentally. The VHDL implementation is tested and verified both through simulation and on hardware. Xilinx is used for logic simulation. A simulation waveform is used to verify the correctness of the implementation, by checking the produced results

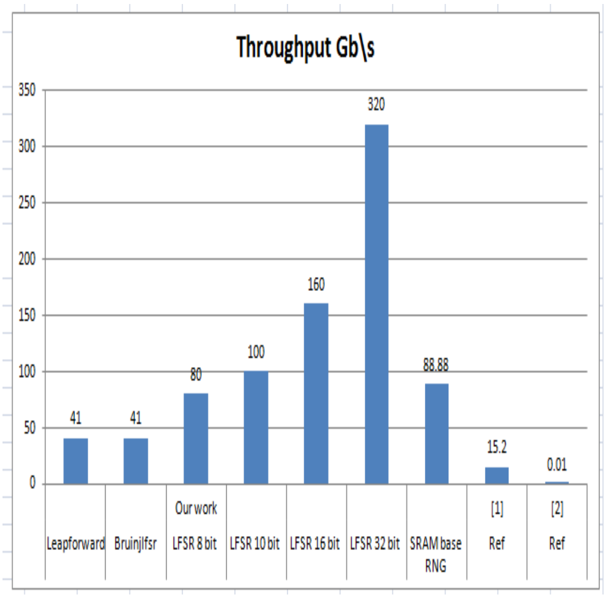
Synthesis report:

COMPONENTS USED	QUANTITY
Registers	130
8-bit register	130
Multiplexers	2
8-bit 64-to-1 multiplexer	2
Xors	1
8-bit xor2	1
Registers	1040
Flip-Flops	1040
Multiplexers	2
8-bit 64-to-1 multiplexer	2
Xors	
8-bit xor2	1

The synthesis report reveals that the gate count of design comprises of 130 flipflops, \two 64X1 multiplexers and 1040 flip flops for SRAM cell design.

Comparative analysis

Cell design	Throughput Gb\s
Leapforward	41
Bruinjlfsr	41
LFSR 8 bit	80
LFSR 10 bit	100
LFSR 16 bit	160
LFSR 32 bit	320
SRAM base RNG	88.88
Base paper	15.2
Base paper	0.01



Cell design	Compuatational Time
Leapforward	0.1
Bruinjlfsr	0.1
LFSR 8 bit	0.1
LFSR 10 bit	0.1
LFSR 16 bit	0.1
LFSR 32 bit	0.1
SRAM base RNG	0.090009001
Base paper	0.002105263
Base paper	0.2

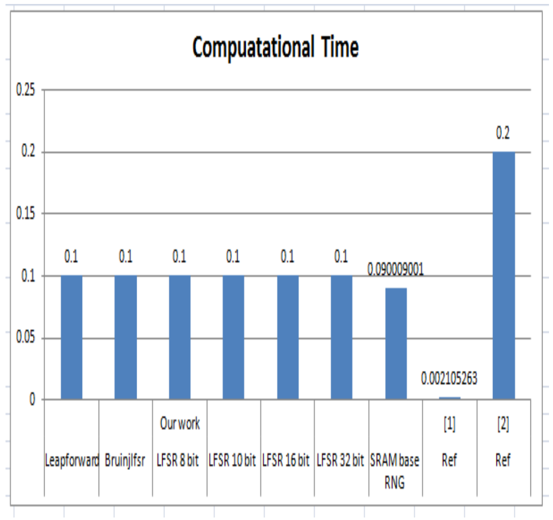


Table1 and 2 shows the comparison of a number of FPGA-based PRNGs, in terms of quality metric, resource usage, and performance. , although consuming more logic resource (this is because of the algorithm complexity), the our random number generator achieves a bit higher throughput. In addition, it is worth to note that the resource usage is less.

7. CONCLUSION

The work is done for the random number generation using eight, sixteen and thirty two bit linear feedback shift register. The use of LFSR will raise the shift register bits and raise the shift rergister length. The observation shows the comparison of a number of PRNG, in terms of computational time and throughput. The code for implementing the required PRBS is designed by programming VHDL code. A 16-bit PRBS is implememted by shifting the input through the D-flip flops and feed backing the outputs of some registers known as taps again into the first register after passing them through a XOR gate. The process of designing LFSR is carried out by first developing the VHDL code for a D-flip flop. The same D- flip flop code is then called 16 times in the main program code to realize the required LFSR.

8. FUTURE SCOPE

- In future the design random number generator using LFSR is use in bit error rate testing for communication data.
- In this current work in the thesis showed the effectiveness of the design. A more involved approach is the overall layout implementation with tests. The parametric optimization of static and dynamic power dissipation for LFSR is possible to reduce by designing the CMOS layout using deep sub micron technology

9. REFERENCES

[1] Yuan Li, Paul Chow, Jiang Jiang, Minxuan Zhang, and Shaojun Wei "Software/Hardware Parallel Long-Period Random Number Generation Framework Based on the WELL Method" Ieee Transactions On Very Large Scale Integration (Vlsi) Systems year 2013.

[2] David B. Thomas, and Wayne Luk "The LUT-SR Family of Uniform Random Number Generators for FPGA Architectures " IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, April 2013.

[3] Piotr Zbigniew Wieczorek, and Krzysztof Gołofit "Dual-Metastability Time-Competitive True Random Number Generator" Ieee Transactions On Circuits And Systems. ,year 2013.

[4] Sharmitha.E.K, Sharmitha.E.K, Nisha Angeline. M, Palanisamy.C "High Throughput LFSR Design for BCH Encoder using Sample Period Reduction Technique for MLC NAND based Flash Memories " International Journal of Computer Applications (0975 – 8887), March 2013.

[5] Srinivasan Krishnaswamy and Harish K. Pillai "On the Number of Linear Feedback Shift Registers With a Special Structure" IEEE Transactions On Information Theory, March 2012.

[6] François Arnault, Thierry Berger, Marine Minier, and Benjamin Pousse " Revisiting LFSRs for Cryptographic Applications " IEEE Transactions On Information Theory, December 2011.

[7] Nilanjan Mukherjee and Janusz Rajski, Grzegorz Mrugalski and Artur Pogiel "Ring Generator: An Ultimate Linear Feedback Shift Register" IEEE Computer Society year 2011.

[8] Fabio Pareschi, Gianluca Setti, and Riccardo Rovatti " Implementation and Testing of High-Speed CMOS True Random Number Generators Based on Chaotic Systems" IEEE Transactions On Circuits And Systems, DECEMBER 2010.

[9] Kiran George and Chien-In Henry Chen "Logic Built-In Self-Test for Core-Based Designs on System-on-a-Chip" IEEE Transactions On Instrumentation And Measurement, May 2009.

[10] Enes Pasalic "On Guess and Determine Cryptanalysis of LFSR-Based Stream Ciphers" IEEE Transactions On Information Theory, July 2009

[11] Enes Pasalic "On Guess and Determine Cryptanalysis of LFSR-Based Stream Ciphers" IEEE Transactions On Information Theory , July 2009.

[12] Walter Aloisi and Rosario Mita "Gated-Clock Design of Linear-Feedback Shift Registers" Ieee Transactions on Circuits and Systems—II: Express Briefs, year june 2008.

[13] Doshi N. A., Dhobale S. B., and Kakade S. R. "LFSR Counter Implementation in CMOS VLSI" World Academy of Science, Engineering and Technology 48 year 2008.

[14] Hong-Sik Kim and Sungho Kang "Increasing Encoding Efficiency of LFSR Reseeding-Based Test Compression" EEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, May 2006.

[15] Rajendra S. Katti, Xiaoyu Ruan, and Hareesh Khattri "Multiple-Output Low-Power Linear Feedback Shift Register Design" IEEE Transactions On Circuits And Systems, July 2006

[16] P. J Coleman-Smith "Verification of the GREAT Total Data Readout System Using Pseudo Random Pattern Generator" IEEE Transactions On Nuclear Science, August 2006.

[17] Timothy O. Dickson, Ekaterina Laskin, Imran Khalid, Rudy Beerkens, Jingqiong Xie, Member, Boris Karajica, and Sorin P. Voinigescu "An 80-Gb/s 231 1 Pseudorandom Binary Sequence Generator in SiGe BiCMOS Technology" IEEE Journal Of Solid-State Circuits,December 2005.

[18] Hong-Sik Kim, Yongjoon Kim,and Sungho Kang " Test-Decompression Mechanism Using a Variable-Length Multiple-Polynomial LFSR" IEEE Transactions On Very Large Scale Integration (Vlsi) Systems,August 2003

[19] Nicholas Kolokotronis and Nicholas Kalouptsidis "On the Linear Complexity of Nonlinearly Filtered PN-Sequences" IEEE Transactions On Information Theory, November 2003.

[20] Toni Stojanovski and Ljupˇco Kocarev "Chaos-Based Random Number Generators—Part I: Analysis" IEEE Transactions On Circuits And Systems—I: Fundamental Theory And Applications, March 2001.

[21] Te Sun Han, Fellow, and Osamu Uchida " Source Code with Cost as a Nonuniform Random Number Generator" IEEE Transactions On Information Theory, March 2000.

[22] Karthik Visweswariah, Sanjeev R. Kulkarni, and Sergio Verd´u "Source Codes as Random Number Generators" IEEE Transactions On Information Theory, March 1998.