Design and Analysis of FPGA based cryptographic N-bit parallel LFSR

Shiv Dutta Mishra
Research Scholar ME (VLSI Design),
Shri Shankaracharya Group of Institutions, Junwani, Bhilai, (C.G.).
INDIA

Prof. Anurag Shrivastav
Department of Electronics and Telecommunication, Faculty of Engineering & Technology,
Shri Shankaracharya Group of Institutions, Junwani, Bhilai, (C.G.).
INDIA

Abstract: The FPGA based implementation of N-bit parallel LFSR Pseudo Random Sequence generator is presented in this paper that can be used in cryptography for securing the Internet traffic in places where low memory utilization and low level of security is required. Random numbers form the centerpiece of cryptography provided the seed that the random number generator provides remains secretive and a high degree of randomness is maintained. FPGA is especially popular for prototyping integrated circuit designs and to develop and simulate a sophisticated digital circuit, realize it on a prototyping device, and verify the operation of its physical implementation. As these Technologies have become accepted mainstream practice so that it is possible to use a PC and an FPGA prototyping board to construct a digital circuit. The concept of using software to design hardware that is controlled by software is implemented. The design of parallel LFSR circuit was implemented in a Vertex 5 series (XCVLX5110T) target device with the use of Verilog as the hardware description language.

Keywords: LFSR, FPGA, PRNG, Cryptography and Verilog

I. INTRODUCTION

The construction of pseudo-random number generator (PRNG) is based on linear feedback shift register (LFSR). This class of generators can be very effectively implemented in hardware, is capable to generate very long pseudorandom sequences with a high-quality statistical distribution.

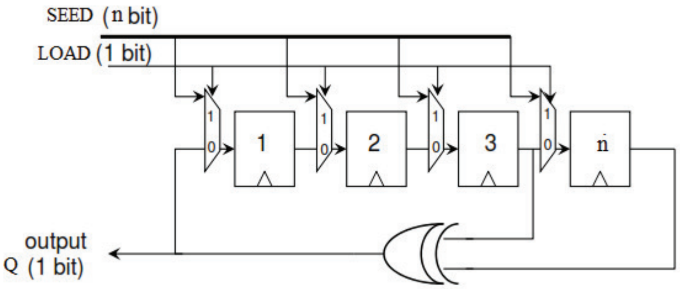

Figure 1. Linear feedback shift register (LFSR)

An n-stage maximum length linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state is shown in fig (1). The only linear function of single bits is XOR, thus it is a shift register whose input bit is driven by the exclusive-or of some bits of the overall shift register value. 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. The arrangement of taps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod 2. This means that the coefficients of the polynomial must be 1's or 0's. This is called the feedback polynomial or characteristic polynomial. For example, in 4 bit LFSR if the taps are at the 4th and 3rd bits, then the feedback polynomial is x4 +x3 +1.Flip-flops are clocked in every clock cycle and only one bit of information is generated per clock cycle. The output can be taken from input or output of any flip-flop.

The parallel architecture of an LFSR designed by M. Lowy [1] is shown in fig (2).Individual cell in the register do not change value except when updated once every N clock cycle instead i.e. only one flip-flop is updated every clock cycle and instead of the bits moving from left to right, the tap of XOR tree move from right to left in the direction of preceding cell. It reduces dynamic power consumption and can generate more than one bit output per clock cycle. The M. Lowy architecture uses an

1. N-phase generator that generates N signal to clock flip flop and is realized by johnson counter of length N/2.
2. Control unit that control the operation of M+N switches and is realized by M+N OR gate.

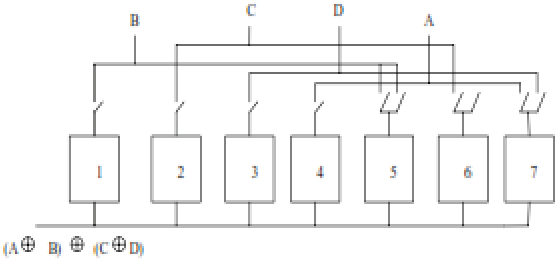

Figure 2. Single output realization of F(X) = 1+ X4 + X7.

M. E. Hamid and C. I. H. Chen[2] proposed a new form of polynomial with two coefficients having the following format:

$$F(X) = 1 + X^{n/2} + X^n$$

Where n is the order of the polynomial shown in fig (3). The proposed polynomial reduces the number of switches required as well as a 3% increase in number of distinct patterns generated.

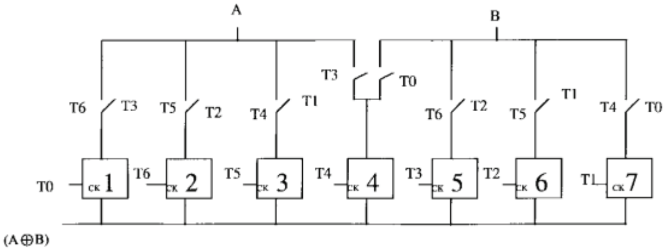

Figure 3. Single output realization of F(X) = 1+ X4 + X7.

Abdullah Mamun and Rajendra Katti [3] proposed design is shown in fig (4) in which additional XOR gates are used and they are permanently connected to the respective flip-flops where as in both Lowy's design and Hamid's design, taps move. It is very simple in design, reduces power consumption significantly, and eliminates the control unit.

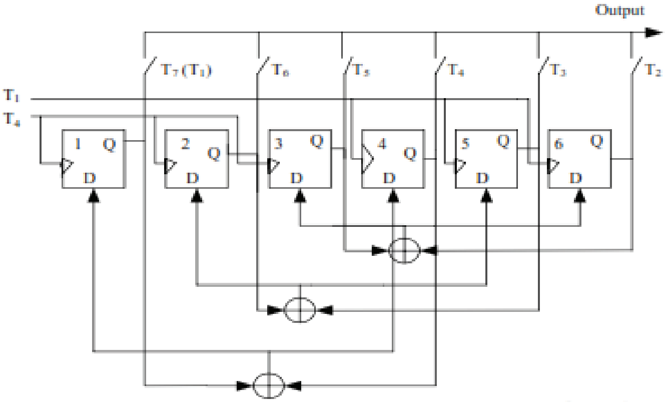

Figure 4. Single output realization of F(X) = 1+ X3 + X6.

Consider polynomial 1+ x2 + x5 for architecture proposed by M. Lowy for obtaining multiple output in a single clock cycle. The control signal Ti (i = 1, 2 ... N), which is used to connect the shift register to the output tap, is a sequentially occurring waveform. The control signal is high for only i mod N clock cycle, where N is the length of the LFSR. The operations (5,2) and (4, 1) are performed at T1, where (x, y) denotes XOR operation of x and y. The value of the XOR operation is the output of the LFSR, obtained at A and B respectively. The outputs A and B are feedback and stored in flip-flop 5 and flip-flop 4 respectively at T2 cycle. At T2, operations (3, 5) and (2, 4) are performed. These values are stored in flip-flop 3 and flip-flop 4 respectively. With the multiple output architecture, a polynomial, of the form 1 + xk1 + xk2 + xk3 + ... + x N can generate. k outputs in a single clock cycle. In the case of 1+ x2 + x5, 2 outputs can be obtained in a single clock cycle. In order to produce all the outputs, 5 clock cycles are required [4].

Similarly in case of M. E. Hamid and C. I. H. Chen[2] design. In the case of 1 + x2 + x5, 2 outputs can be obtained in a single clock cycle. In order to produce all the outputs, 5 clock cycles are required.

The same polynomial, 1+x2+x5, implemented in the architecture as proposed by Katti are as shown in Fig. (3). At T1, operations (5, 2) and (4, 1) are performed. Values of these operations are stored in flip- flop 5 and flip-flop 4 respectively at T2. At T2, operations (5, 3) and (4, 2) are performed. The result of the XOR operation is stored in flip-flop 3 and flip-flop 2 respectively at T3. At T3, only one operation (3, 1) is performed, and the result is stored in flip-flop 1 at T1. The operation can be summarized as shown in Table 1.

Table - 1 Update of flip-flops, F(X) = 1+ X3 + X6

Clock cycle	**1**	**2**	**3**	**4**	**5**	**6**
XOR result of	**6,3**	**5,2**	**4,1**	**6,3**	**5,2**	**4,1**
Stored at FF	**6**	**5**	**4**	**3**	**2**	**1**

With the multiple output in this architecture, the numbers of control signals required are N / k1, where N is the length of LFSR and k1 is the number of simultaneous outputs. In this example, 3 control signals are needed, and all the outputs are produced in 3 clock cycle [4].

In the proposed design a single XOR gate is used and is permanently connected to the respective flip-flops with bits moving LSB to MSB flip flop and position of taps are given using M. E. Hamid and C. I. H. Chen polynomial produces N bit parallel output in a single clock.

This paper examines the full procedure of implementing N-bit parallel LFSR using a high-level hardware description language; Verilog, combined with the usage of FPGA technology. In section II we have proposed design of N Bit parallel LRSR and in Section III the complete design was synthesized for FPGA devices Virtex 5. The design is implemented and verified on a Virtex 5 FPGA development board from Xilinx using device XCVLX5110T and then simulated using model sim ISE simulator. In section V conclusion is given.

II. Proposed N-bit parallel LFSR

The circuit diagram of proposed N bit parallel LFSR is shown in fig 5.

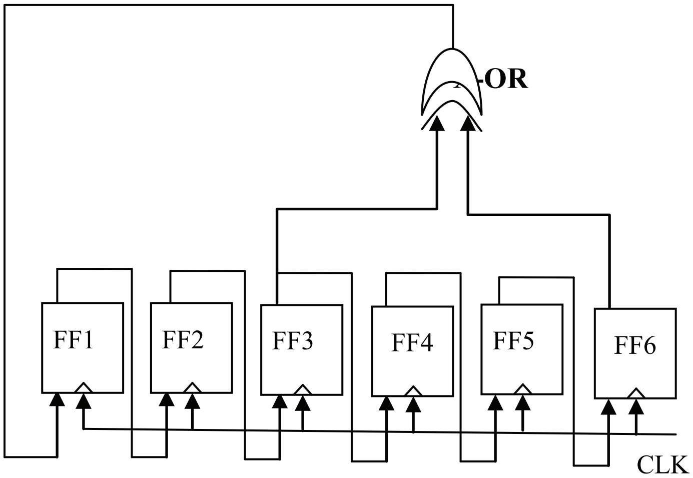

Figure 5. Output realization of F(X) = 1+ X3 + X6.

The designing of proposed design is done by the help of N flip flop, the flip flop used is D flip flop and a XOR gate. Output is taken from each flip flop with bits moving LSB to MSB flip flop in each clock pulse, except the LSB flip flop whose input is output of XOR of N/2 and N flip flop. XOR gate is permanently connected to the respective flip-flops of polynomial $F(X) = 1 + X^{n/2} + X^n$.
Let us consider the polynomial for N=6, F(X) = 1+ X3 + X6. Table 3 shows the storing procedure in order to obtain the parallel implementation. In cycle 1, XOR result of flip-flops 3 and 6 (which is calculated in the previous cycle) is stored in flip-flop 1. Similarly in clock cycle 2, 3, 4, 5, and 6 XOR results of flip- flops (2,5); (1,4); (updated FF 1,3); (updated FF 2,2); and (updated FF 3,1) should be stored in flip- flop 3,4,5 and 6 respectively. The operation can be summarized as shown in Table 2.

Table 2 Update of flip-flops, F(X) = 1+ X3 + X6

Clock cycle	**1**	**2**	**3**	**4**	**5**	**6**
XOR result of	**6,3**	**5,2**	**4,1**	**3, updated FF 1**	**2, updated FF 2**	**1, updated FF 3**
Stored at FF	**1**	**2**	**3**	**4**	**5**	**6**

III. Experiment and Result

The proposed design is modeled in Verilog [5] [6] [7] [8] [9] using parameter n=6; input R[n-1:0], load and Clk the output is Q[n-1:0]. When load pin is high value of R is given to Q,To specify repetitive structure of LFSR for loop is used. FOR statement is procedural statement and is placed inside the always block. In our code for loop contains two statements. These statements define shifting of values of Flip Flop from LSB to MSB and XOR operation. These signals retain these values until it is again re-evaluated by positive edge of next clock pulse.

From table 1 and 2 it is found that the values updated at flip flop is same except the order in which they are stored in our design the values are stored in flip flop 1,2,3,4,5,6 respectively whereas in Abdullah Mamun and Rajendra Katti architecture values are stored in flip flop 6,5,4,3,2,1 respectively.

A. SYNTHESIS

The complete experimental work start with the proposed design is modeled in Verilog. Functional simulation, schematic generation, RTL generation, synthesis for hardware platforms in FPGA was done and finally implemented in specific target hardware to verify the proper functionality of the design using Xilinx ISE Design Suite 10.1i. the RTL Schematic and Technology Schematic of N bit parallel LFSR having polynomial 1+ X3 + X6 is shown in fig (6) and (7) respectively. HDL Synthesis of N bit parallel LFSR is given below

```
Synthesizing Unit <shiftn>.
    Related source file is "shiftn.v".
    Found 6-bit register for signal <Q>.
    Found 1-bit xor2 for signal <Q_0$xor0000> .
    Summary:
            inferred   6 D-type flip-flop(s).
            inferred   1 Xor(s).
```

Which is correct as our design uses 6 D-type flip-flop and 1XOR.

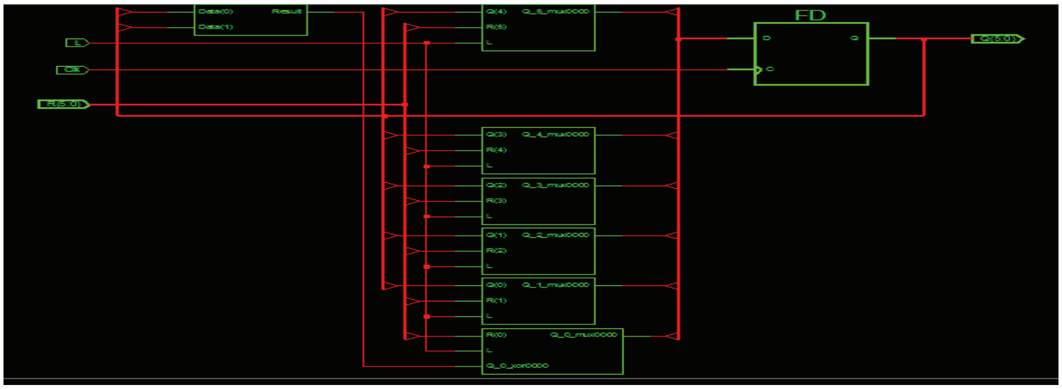

Figure 6 RTL Schematic of N bit parallel LFSR having polynomial 1+ X3 + X6

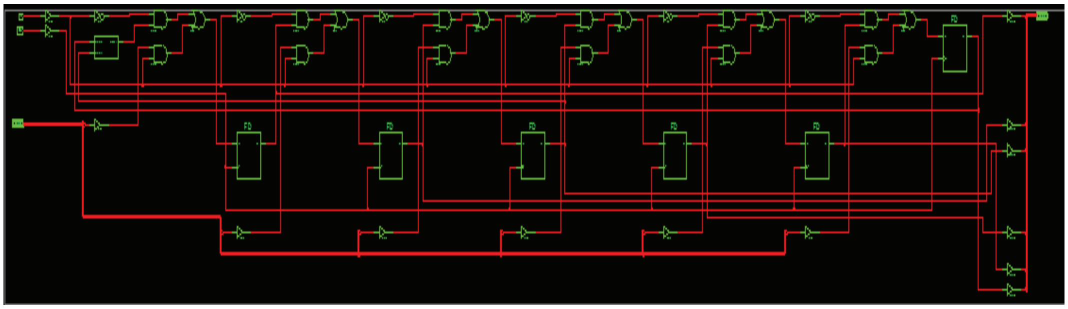

Figure 7 Technology Schematic of N bit parallel LFSR having polynomial 1+ X3 + X6

B. *Simulation Timing Waveform*

Timing simulation waveform of N bit parallel LFSR having polynomial 1+ X3 + X6 and SEED = "001011" is shown in Fig. (8).From the simulation waveform we can observe 9 random output states are coming there after it is repeating each of 6 bit 001011, 010110, 101101, 011010, 110100, 101000, 010001, 100010, 000101.

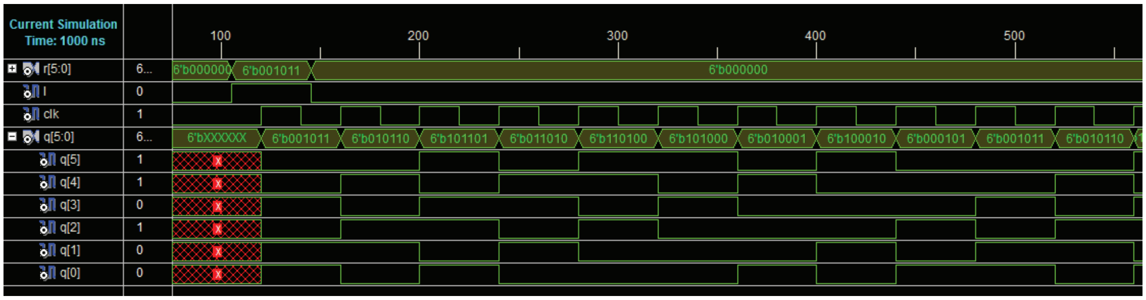

Figure 8. Simulation waveform of N bit parallel LFSR using SEED = "001011"

IV.CONCLUSION

In M. Lowy[1] and M. E. Hamid and C. I. H. Chen[2] [4] design , 2 output of 1 bit can be obtained in a single clock cycle with increase in number of switches and in Abdullah Mamun and Rajendra Katti[3] [4] 2 output of N/2 bit can be obtained in a single clock cycle with reduce dynamic power consumption. This paper proposes an N-bit parallel LFSR which produces N-bit output in a single clock cycle, eliminating phase generator and control circuit but the power dissipation is increased as complete structure is clocked every clock cycle. Since all the flip flops are clocked therefore it is advantageous to take output from the entire flip flop. The number of output state is 9 where as for normal 6 bit shift register the number of output state are 6 only then it starts repeating again. Total memory usage is 150636 kilobytes which is very low. So it can be used in cryptography where low memory utilization and low level of security is required.

REFERENCES

[1] M. Lowy, “Parallel implementation of linear feedback shift registers for low power applications,” IEEE Trans. Circuits Syst. II, vol. 43, pp. 458–466, June 1996.

[2] Hamid, Muhammad and E. Chen, Chien-In Henry, “Note to low- power linear feedback shift registers,” IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, Issue 9, pp 1304-1307, September 1998.

[3] Abdullah Mamun and Rajendra Katti. “A new parallel architecture for low power linear feedback shift registers”

[4] Shilesh Malliyoor and Chao You “Comparison of hardware implementation and power consumption of low-power multiple output linear feedback shift register”journal of engineering, computing and architecture. Volume 1, Issue 1, 2007

[5] Field programmable gate array (2011). Wikipedia website. Online Available:http ://en.wikipedia.org /wiki/ Field-programmable_gate_array.

[6] Stephen Brown, Zvonko Vranesic”fundamental of Digital logic with Verilog design,2e, Tata McGraw-Hilla,delhi,2007.

[7] Bhasker J, “A VHDL Primer”, P T R Prentice Hall, Pages 1-2, 4-13, 28-30.

[8] http://www.xilinx.com/literature .

[9] http://www.xilinx.com/support .