DESIGN OF HIGH-SPEED VLSI ARCHITECTURE FOR LFSR WITH MAXIMUM LENGTH FEEDBACK POLYNOMIAL

Ms. Meenal R. Dadhe
PG Scholar,
GHRAET, RTMN University,
Nagpur, India

Prof. Sanjay Tembhurne
Electronics and Communication Engineering,
GHRAET, RTMN University,
Nagpur, India

Abstract— The main purpose of high-speed architecture of linear feedback shift register (LFSR) based on PN Sequence generator technique. It is used for various cryptography application and for designing encoder, decoder in different communication channel. Depend on the feedback polynomial total number of random sequence generator on LFSR. It is simple counter so its count maximum of 2^{n-1} by using maximum feedback polynomial. Here in this work we implement LFSR by using VHDL to study the performance and analysis the behaviour of randomness .The analysis is conceded out to find number of gates, memory and speed requirement as the number of bits is increased. We proposed LFSR architecture based on serial, parallel, combined parallel and pipelining algorithm to minimize delay of the system.

***Keywords** - Linear Feedback Shift Register, VHDL.*

I. INTRODUCTION

For generating data encryption keys, random numbers are very useful in the various applications such as communication channel, bank security. it is also used to design encoder and decoder for sending and receiving data in noisy communication channel. When discussing single numbers. Random number generator is a computational device to generate a sequence of numbers or that lack any pattern. There are various methods for pseudo-random numbers are known. Most of them are based, on linear congruential equations and require a number of time consuming arithmetic operations. In contrast, the use of feedback shift registers permits very fast generation of binary sequences. Shift register sequences of maximum length (m-sequences) are well suited to simulate truly random binary sequences. As we change the feedback polynomial the run-length as well randomness also changes [1].

II. 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. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. [8]. the initial value of the LFSR is called the seed. Because the register has a finite number of possible states. it must 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.

III. RANDOM SEQUENCE GENERATOR

The PN Sequence Generator block generates a sequence of pseudo random binary numbers using a linear-feedback shift register (LFSR). This block implements LFSR using a simple shift register generator configuration. A pseudo noise sequence can be used in a pseudo random scrambler and descrambler. It can also be used in a direct-sequence spread-spectrum system. The PN Sequence Generator block uses a shift register to generate sequences, as shown in fig below.

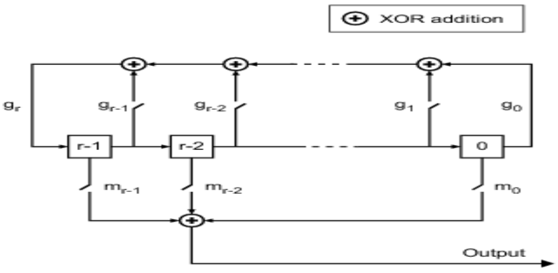

Fig 1 : Random Sequence Generator using LFSR

All registers in the generator update their values at each time step, according to the value of the incoming arrow to the shift register. The adders perform addition modulo 2. The shift register is described by the Generator Polynomial parameter, which is a primitive binary polynomial in *z*, $g_r z^r + g_{r-1} z^{r-1} + g_{r-2} z^{r-2} + ... + g_0$. The coefficient g_k is 1 if there is a connection from the kth register, as labeled in the preceding diagram, to the adder. The leading term g_r and the constant term g_0 of the Generator Polynomial parameter must be 1 because the polynomial must be primitive. For example, `[1 0 0 0 0 0 1 0 1]` and `[8 2 0]` represent the same polynomial, $p(z) = z^8 + z^2 + 1$. The Initial states parameter is a vector specifying the initial values of the registers. The Initial states parameter must satisfy these criteria:

- All elements of the Initial states vector must be binary numbers.
- The length of the Initial states vector must equal the degree of the generator polynomial [7]

IV. LFSR ARCHITECTURE

4.1 Piplining

In a pipelined system:

- Pipelining can be employed to achieve higher speed in feedback loops.
- In an M-level pipelined system, the number of delay elements in any path from input to output is (M-1) greater than that in the same path in the original sequential circuit
- Pipelining reduces the critical path, but leads to a penalty in terms of an increased latency
- Latency: the difference in the availability of the first output data in the pipelined system and the sequential system
- Two main drawbacks: increase in the number of latches and in system

4.2 Parallel Architecture

Parallel processing and pipelining techniques are duals each other: if a computation can be pipelined, it can also be processed in parallel. Both of them exploit concurrency available in the computation in different ways.

- Parallel processing system is also called block processing, and the number of inputs processed in a clock cycle is referred to as the block size
- In this parallel processing system, at the k-th clock cycle, 3 inputs x(3k) x(3k+1) and x(3K+2) are processed and 3 samples y(3k), y(3k+1) and y(3k+2) are generated at the output

• Note 1: In the MIMO structure, placing a latch at any line produces an effective delay of L clock cycles at the sample rate (L: the block size).

V. LITERATURE REVIEW

An exhaustive literature review has been carried out related to the work to find out the current research. Abstracts of some of most relevant research works are reported in the following paragraph

GU Xiao-chen, ZHANG Min-xuan, "Uniform Random Number Genrator using Leap-Ahead LFSR Architecture" 2009 International Conference on Computer and Communications Security, 2009 IEEE

This paper, introduce a new kind of URNG using Leap-Ahead LFSR architecture which could generate an m-bits random number per cycle using only one LFSR. the Leap-Ahead LFSR Architecture URNG consumes less than 40 slices which is only 10% of what the Multi-LFSRs architecture consumes and acquires very good Area Time performance and Throughput performance that are 2.18×10-9 slices×sec per bit and 17.87×109 bits per sec. the Leap-Ahead architecture works slower than the Multi-LFSR architecture. This is because the feedback logic is much more complicated in the Leap-Ahead architecture. And the Fan-Out of each register is larger, too. This drawback is much obvious in Galois type architecture, because not only the Fan-Out increases but also the logic stages of the feedback circuit increases from 1 to 4. So, the working frequency decreases from 1146 MHz to 993 MHz.

Chan-Bok Jeong and Dae-Ho Kim, "High-Speed Architecture for k-dimensional LFSR in H/W Implementation" Electronics and Telecommunication Research Institute Daejeon, Korea 2011 IEEE

The analysis of the proposed LFSR architecture demonstrates that the proposed k-dimensional LFSR architecture is k times as fast as a conventional LFSR architecture and the used processing time for scrambling is enough to implement scramble function for high-speed applications such as L TE-Advanced. The architecture of k bits-in and k bits-out is called k-dimension as the proposed LFSR structure. It is possible to transform the 1-dimensional LFSR architecture into the k-dimensional LFSR architecture.

Amit Kumar Panda*, Praveena Rajput, Bhawna Shukla, "FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial using VHDL", 2012 International Conference on Communication Systems and Network Technologies.

LFSR based PN Sequence Generator technique is used for various cryptography applications and for designing encoder, decoder in different communication channel. It is more

Important to test and verify by implementing on any hardware for getting better efficient result. The total number of random state generated on LFSR depends on the feedback polynomial. As it is simple counter so it can count maximum of 2^n -1 by using maximum feedback polynomial. Here in this paper implemented 8, 16 and 32-bit LFSR on FPGA by using VHDL to study the performance and analysis the behavior of

Performance	8 bit	16 bit	32 bit
Time to complete the total states	40ns to5140ns=5 100ns	20ns to1310720ns =1310.7us	20ns to 85899345920ns= 85.9ns
Total number of random state generating	255	65535	429,49,6,295
Shift Register	08	16	32
Xor gate	01	01	01
Number of slices	04	09	18
No. of slice Flip-Flop	08	16	32
No. of 4 i/p LUT	01	01	01
Total memory usage	185904kb	185904kb	185904kb
GCLK	01	01	01
DELAY	7.27ns	7.27ns	7.27ns
Total pin	10	18	34

randomness. The analysis is conceded out to find number of gates, memory and speed requirement in FPGA as the number of bits is increased. In this paper delay in 8bit, 16bit, and 32bit LFSR is7.271ns and number of slices is 4, 9, 18[7].

Hao Chen, "CRT-based high-speed parallel architecture for long bch encoding" ieee transactions on circuits and systems—ii: express briefs, vol. 56, no. 8, august 2009

Bose–Chaudhuri–Hocquenghen (BCH) error correcting codes are now widely used in communication system and digital technology. The direct linear feedback shifted register (LFSR)-based encoding of a long BCH code suffers from the large fan-out effect of some XOR gates. This makes the LFSR-based encoders of long BCH codes not keep up with the data transmission speed in some applications. The technique for eliminating the large fan-out effect by J-unfolding method and some algebraic manipulation has been proposed. In this brief, present a Chinese remainder theorem (CRT)- this paper architecture can be used to eliminate the fan-out bottleneck. [4].

E.Khushboo Sewak, Praveena Rajput and Amit Kumar Panda*," FPGA Implementation of 16 bit BBS and LFSR PN Sequence Generator: A Comparative Study"2012 IEEE Students' Conference on Electrical, Electronics and Computer Science

The main purpose of this paper is to study the FPGA implementation of two 16 bit PN sequence generator namely Linear Feedback Shift Register (LFSR) and Blum-Blum-Shub (BBS). In this paper used FPGA to explain how FPGA's ease the hardware implementation part of communication systems. The logic of PN Sequence Generator presented here can be changed any time by changing the seed in LFSR or by changing the key used in BBS. The analysis is conceded out to find number of gates, memory and speed requirement in FPGA for the two methods. In this paper CPU time is taking for 4-Bit LFSR is 0.39 sec where as for BBS it is 4.11 sec. Memory utilization is more in BBS than LFSR as the number of FFs used in BBS are more[6].

V. CONCLUSION

This paper presented to minimize the delay of LFSR architecture used to generate random PN sequence and to generate maximum length PN Sequence with effective amount gates .LFSR is used in various applications and in various architecture for increasing the speed of operation.

References

[1] Proakis, John G., Digital Communications, Third edition, New York, McGraw Hill, 1995.

[2] Golomb, S.W., Shift Register Sequences, Aegean Park Press, 1967.

[3] Jui-Chieh Lin, Sao-Jie Chen, Senior Member, IEEE, and Yu Hen Hu, Fellow, IEEE, "Cycle-Efficient LFSR Implementation on Word-Based Microarchitecture" IEEE transactions on computers, vol. 62, no. 4, april 2013

[4] Hao Chen," CRT-Based High-Speed Parallel Architecture for Long BCH Encoding" IEEE transactions on circuits and systems—ii: express briefs, vol. 56, no. 8, august 2009

[5] Manohar Ayinala , Keshab K. Parhi," High-Speed Parallel Architectures for Linear Feedback Shift Register" IEEE transactions on signal processing, vol. 59, no. 9, september 2011

[6] Khushboo Sewak , Praveena Rajput, Amit Kumar Panda," FPGA Implementation of 16 bit BBS and LFSR PN Sequence generator" 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science.

[7] Amit Kumar Panda, Praveena Rajput, Bhawna Shukla," FPGA Implementation of 8,16 and 32 Bit LFSR with Maximum Length Feedback Polynomial Using VHDL" 2012 International Conference on Communication Systems and Network Technologies.

[8] C. Cheng and K. K. Parhi, "High speed parallel CRC implementation based on unfolding, pipelining, retiming," IEEE Trans. Circuits Syst.II, Expr. Briefs vol. 53, no. 10, pp. 1017–1021, Oct. 2006.

[9] 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.