GENERATION OF PSEUDO-RANDOM NUMBER BY USING WELL AND RESEEDING METHOD

V.Divya Bharathi[1], Arivasanth.M[2]

[1]PG Scholar, M.E-VLSI Design,Srinivasan Engineering College, Perambalur, TamilNadu, India.

[2]Assistant Professor, Department of ECE, Srinivasan Engineering College, Perambalur, TamilNadu, India.

***Abstract*- This paper presents a hardware architecture for the generation of Pseudo-Random number using Well Equi-Distributed Long-Period Linear (WELL) Generator and RESEEDING method. A random number is generated by using WELL method first and its performance was analyzed. When the random numbers are generated, the repeating patterns will occur. For avoiding the repeating pattern we use RESEEDING method. Here the non-repeating patterns are obtained and its speed get increased to 30%.**

***Index Terms* -WELL, PRNG, RESEEDING method.**

Manuscript received on March , 2015.

V.Divya Bharathi, *ME(VLSI Design), Srinivasan Engineering college, perambalur, India.*

Arivasanth.M, Assistant Professor, Srinivasan Engineering college, Perambalur, India.

I INTRODUCTION

Random numbers are frequently used in scientific applications, particularly for simulations [1]. In Monte-carlo method, the simulation result generates the poor quality of random numbers. Therefore high quality random numbers are great importance to many scientific applications [1]. Pseudo-Random Number Generators (PRNG) is generally adopted in such simulations because of their performance and reproducibility [3], but the quality of PRNG is not sufficient for certain simulations. Mersenne twister (MT) is a PRNG that is highly suited to simulations for uniformity, long period and good equi-distribution. One main drawback of Mersenne Twister is that sensitivity of poor initialization and takes long time to recover from zero excess initial state [1]. In order to solve this problem WELL generator is used. The period of this WELL generator will be 2^{19937}-1. The

ISSN: 2278 – 909X

random number generator can be used in cryptography applications, when the seed is in secret. Same kind of random numbers can be generated by sender and receiver for keys. The FPGA-optimized generators cannot be used in practical applications because; the process of constructing the generator will be time–consuming. Advances in VLSI include instantiating RNG to meet their needs of specific application. In this section, types of random number generator are briefly explained.

a)Pseudo Random Number Generator:

A pseudo-random number generator (PRNG) uses computational algorithms. Long sequences are produced by using PRNG with approximate results. A key or Seed is given as the initial value for PRNG. Depending on the quality of random number generator, it will be cryptographically secure i.e., if the output is predictable. Some of the most commonly used PRNG is the Linear Congreuntial Generator (LCG).After being initialized with a seed value the internal state of the generator completely determines the next bit to be generated [2]. Generally PRNG is used to produce much longer sequence that appears to be random.

b)True Random Number Generator:

True random number generator (TRNG) produces random numbers by measuring the physical phenomena. Example for TRNG will be atmospheric noise, thermal noise etc; In TRNG, the output is based only on the physical process and not on any previously produced bits [2].

II WELL GENERATOR

Block diagram:

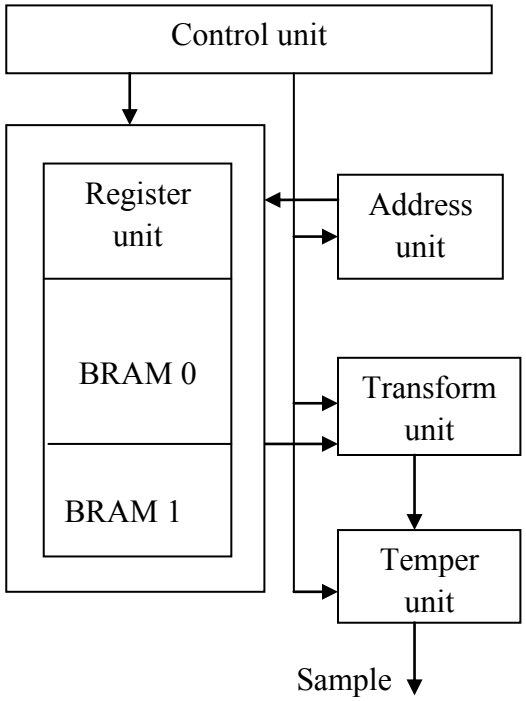

Fig.1 Block Diagram of WELL Method

The existing WELL method consists of Block RAM (BRAM) which is the dedicated two port memory that consists of several kilobits of RAM.BRAM performs 6 read and 2 write operation,but it is reduced to 4 read and 2 write operation by using read before write operation and buffer [1]. This

ISSN: 2278 – 909X

will achieve one sample per cycle throughput. The transform and temper unit performs temper and transform operations and the output produced as "sample".

***WELL generator output**:*

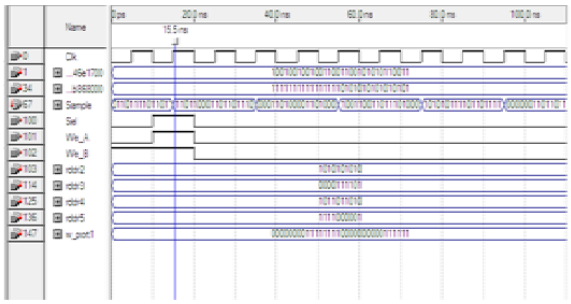

Fig.2 WELL Generator Output

In BRAM, when WE is enable, 32-bit input is given and write operation is performed. When WE become low, the read operation is performed and therefore the output will be displayed at Dout. The output Z4 (32-bit) value is obtained by performing various shift and XOR operations on the read and write ports according to the architecture given in transform unit. The output of the temper unit is sample, which is obtained by shifting and XOR operations of Z4 (output of transform unit) and other Hexadecimal values. Register unit is used to store the given values. i.e., the value of 32-bit w_port1 is provided as the input and the same value is obtained as the output in 32 bit r_port1.By combining the output values of transform unit, temper unit, register unit and BRAM unit, the output of WELL generator is obtained as sample.

***WELL parallel PRNG output**:*

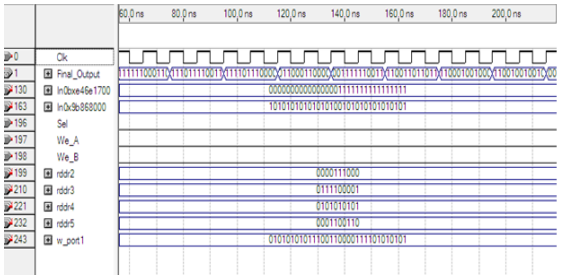

Fig.3 WELL PRNG Output

The sample is declared as a final 32bit output value of a single WELL generator. Therefore the random number is generated as sample for the WELL algorithm. Now, the 32-bit outputs (sample) of 4 WELL PRNGs are arranged in parallel manner to produce the final Pseudo-random Number.

III PROPOSED WORK

In this paper, the WELL generator is used to generate the random numbers and stores the result in BRAM. The LFSR module generates the random number using shifting and XOR operations. The LFSR unit will produce the random patterns by means of the input provided to BRAM and temper unit. Now, the sample provided will be a non-repeating pattern.

BRAM:

BRAM is a Block RAM, which is a two-port memory (PORT A and PORT B) containing

ISSN: 2278 – 909X

several kilobits of RAM. These blocks are denoted as FPGA blocks Each FPGA block consists of small block called Look up table (LUT). It is used for performing logical functions which we can then reconfigure it as few bits of RAM.

LFSR (Linear Feedback Shift Register):

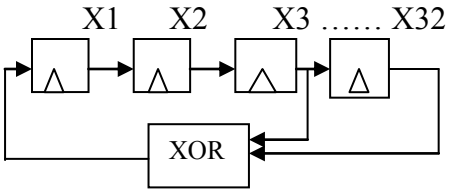

Fig.4 LFSR Block Diagram

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 exclusive-OR (XOR) of some bits of the overall shift register value. When the output of a D flip flop is loaded with a seed value and when the LFSR is clocked, it will generate a PN sequence of 0s and 1s.The period of sequence is 2^n-1,where 'n' is the number of shift registers used in the design. For 32-bit LFSR, the states will be 4294967295.

Advantage of LFSR:

1) Generate large number of random numbers.

2) Secure for cryptography applications.

Block diagram:

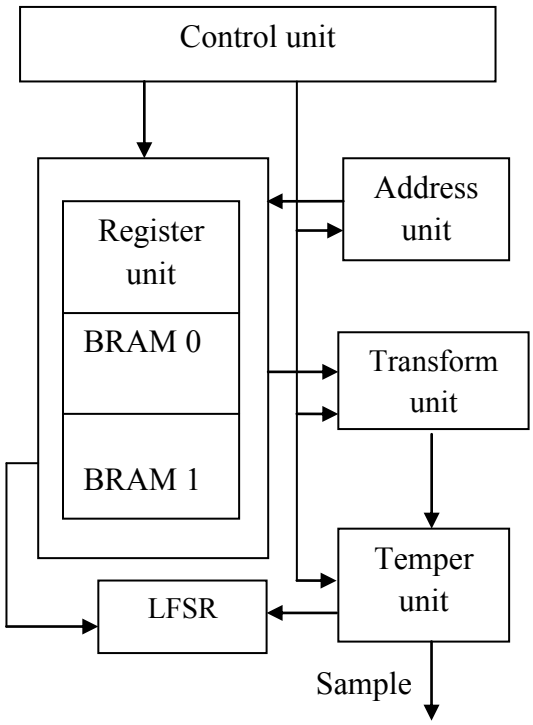

Fig.5 Block Diagram for Proposed Method

The BRAM block is used to store the bits. The BRAM memory will access the random bits and produce the output as DOUTA and DOUTB, since two dual- Port BRAM is used. There will be a separate storage for the values '0' and '1'.BRAM consists of addra, addrb, Din, Dout, WEa and WEb pins. The LFSR unit will produce the random patterns by means of the input provided to BRAM and temper unit. Now, the sample provided will be a non-repeating pattern.

ISSN: 2278 – 909X

32-bit LFSR output:

Fig.6 32-bit LFSR

The 32-bit LFSR will generate 4294967295 random outputs. When rst is high, the input data is provided and the output of LFSR is generated by using 32-D-flip flop.

Proposed Output:

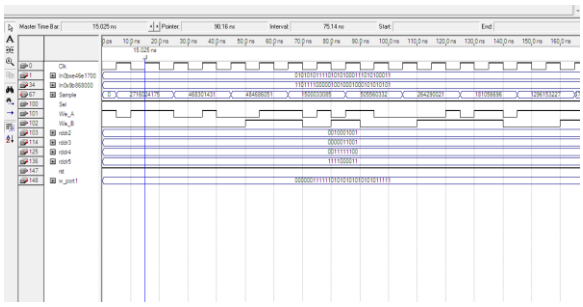

Fig.7 Non-Repeating Pattern

By using LFSR, the Non-Repeating pattern is achieved and the speed is increased by 30%.

IV CONCLUSION

The random patterns are generated by using WELL generator and its performance was analyzed. The speed is enhanced by parallelizing the WELL PRNGs into number of sub streams. This work will produce a sequence of long period, therefore repeating patterns are obtained. To avoid these repeating patterns, RESEEDING method is used. From the simulation results, Non-repeating patterns are obtained. Hence the 30% of the speed get increases.

***References*:**

[1] Yuan Li, Paul Chow, Senior Member, IEEE, 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,VOL.22,NO.5, May2014.

[2] Ms.S.Mukileswari, Mr. M. Sanjeev,”A VLSI Framework for Parallel Long-Period Random Number Generator Based on the Reseeding Method” National Conf on ‘Recent Advances In Communication Engineering”,2014

[3] S. Konuma and S. Ichikawa, ”Design and evaluation of hardware pseudo-random number generator MT19937,” *IEICE Trans. Inf. Syst.*, vol.88, no.12, pp. 2876-2879, Dec.2005.

[4] P. L’Ecuyer and F. Panneton, ”Fast random number generators based on linear recurrences modulo2: Overview and comparison,” in *Proc.37th Conf. Winter Simul.*, 2005, pp. 110-119.

ISSN: 2278 – 909X

[5] M. Matsumoto and T. Nishimura, ”Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator”, ACM Trans. Model. Comput. Simul., vol.8,no.1,pp.3-30, Jan.1998

[6] F. Panneton, P. L’Ecuyer, and M. Matsumoto, “Improved long-period generators based on linear recurrences modulo 2, ”ACM Trans. Math. Softw., vol. 32, no. 1, pp. 1-16, Mar. 2006.

[7] D.B. Thomas and W. Luk, ”High quality uniform random number generation through LUT optimized linear recurrences,” in Proc. IEEE INT. Conf. Field-Program. Technol., Dec.2005, pp. 61-68.

[8] D.B. Thomas and W. Luk, “The LUT-SR family of uniform random number generators for FPGA architectures,” IEEE Trans. Very Large Scale Integrat. (VLSI) Syst., vol. 21, no.4, pp. 761-770, Apr. 2013

First Author

V.Divya Bharathi received the Bachelor’s degree from Dhanalakshmi Srinivasan Engineering College Perambalur, India and doing Masters degree in VLSI Design at Srinivasan Engineering College Perambalur, India.

Second Author

Arivasanth.M received the Bachelor’s degree from Adhiyamaan college Engineering, Hosur, India and done Master’s degree in Thiruvalluvar college of Engineering and Technology ,Vandavasi, India. He is currently working as a Assistant Professor with the Department of Electronics and Communication Engineering, Srinivasan Engineering college, Perambalur.

ISSN: 2278 – 909X