ELectronics EXpress

IEICE Electronics Express, Vol.9, No.17, 1391–1396

Comparison between 2D cellular automata based pseudorandom number generators

Cesar Torres-Huitzil[1a)], **Marco Delgadillo-Escobar**[1], **and Marco Nuno-Maganda**[2]

[1] *CINVESTAV-IPN, Parque Cientifico y Tecnologico TECNOTAM Km. 5.5 carretera Cd. Victoria-Soto La Marina, C.P. 87130, Mexico*

[2] *Universidad Politecnica de Victoria, Parque TECNOTAM Km. 5.5 carretera Cd. Victoria-Soto La Marina, C.P. 87130, Mexico*

a) *ctorres@tamps.cinvestav.mx*

Abstract: Pseudorandom number generators (PRNGs) should satisfy two main criteria, high randomness quality and fast computation of a sequence of numbers. In this paper, a comparative study of two-dimensional Cellular Automata (CA) based PRNGs is performed to evaluate the randomness quality and the hardware constraints involved in terms of configuration parameters such as, transition rules, neighborhoods and bit extraction schemes. Experimental results show that CA-based PRNGs present good randomness quality using standard test suites, and they are well suited for parallel implementations in Field Programmable Gate Array (FPGA) technology taking advantage of the on-chip fine-grain and distributed computational resources.

Keywords: Cellular automata, pseudorandom numbers, FPGA

Classification: Electron devices, circuits, and systems

References

[1] M. Tomassini, M. Sipper, and M. Perrenoud, "On the generation of high-quality random numbers by two-dimensional cellular automata," *IEEE Trans. Comput.*, vol. 40, pp. 1146–1151, 2000.

[2] B. Shackleford, M. Tanaka, R. J. Carter, and G. Snider, "FPGA implementation of neighborhood-of-four cellular automata random number generators," *Proc. 2002 ACM/SIGDA Tenth International Symposium on Field-programmable Gate Arrays*, New York, USA, pp. 106–112, Feb. 2002.

[3] S.-U. Guan, S. Zhang, and M. Quieta, "2-D CA variation with asymmetric neighborship for pseudorandom number generation," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 23, pp. 378–388, 2004.

[4] W.-M. Pang, T.-T. Wong, and P.-A. Heng, "Generating massive high-quality random numbers using GPU," *Proc. IEEE World Congress on Computational Intelligence*, Hong Kong, China, pp. 841–847, June 2008.

[5] S. Das and B. K. Sikdar, "A scalable test structure for multicore chip," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 29, no. 1, pp. 127–137, 2010.

DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012

1 Introduction

PRNGs play an important role in several fields such as large-scale biological simulations, artificial evolutionary computation, cryptography and hardware testing as substitutes of truly random number generators [1]. Each application imposes different statistical and performance requirements to PRNGs but most implementations trade the randomness quality and the associated computational complexity to produce them. Several studies have shown that CA are an alternative technique for PRNG, being one of the main motivations the good randomness and their hardware-friendly nature: CAs are simple, regular, locally interconnected, and modular structures [2]. In this paper, the analysis of two CA-based PRNGs (two-dimensional CA, transition functions, and the number of neighbors) is presented and used to propose a new CA configuration with improved randomness quality. Additionaly, an FPGA-based configurable architecture is used to evaluate performance-quality-cost trade offs for massive parallel implementations of PRNGs.

2 Two-dimensional CA-based PRNG

Two-dimensional (2D) CA-based PRNG consists of an array of locally interconnected cells. Each cell holds a binary cell state which corresponds to a particular bit in the random number. The number of cells used in the grid is adjusted depending on the required wordlength for random numbers. The connectivity among interconnected cells is defined locally in the cell. An m-input boolean transition function is defined for each cell, where m is the number of neighbors used for the cell. Each time a new random number is generated, every cell updates its own state by computing this function at a given time. After all cells update their states, a new random number is obtained by collecting the bits scattered among the cells.

For the purpose of evaluating the quality-performance implementation tradeoff, two CA configurations were used as reference. An homogeneous assymetric connections CA proposed in [2], conceptually shown on the left in Figure 1, presents good randomness quality based on a 2D grid of 8x8 cells with periodic toroid boundary conditions. The connectivity rules for a given cell is as follows $\{2n2w, c, n2e, 2se\}$. Authors used the compass directions n (north), s (south), w (west), and e (east) to indicate unit displacement along columns and rows relative to a center cell c. Several steps in a given direction are indicated by a factor that preced the direction. The 2D CA proposed in [1], shown on the right in Figure 1, is another PRNG that produces high quality sequences of random numbers. Figure 1 shows the so called rules map, indicating different transition rules for cells; missing cells on the boudary are considered to be null. Let S be the state of the cell at position (i, j), at time t, its state at the next time step, is then computed as:

$$S_{i,j}(t+1) = x \oplus (c \cdot S_{i,j}(t)) \oplus (n \cdot S_{i-1,j}(t)) \oplus (w \cdot S_{i,j-1}(t)) \oplus (s \cdot S_{i+1,j}(t)) \oplus (e \cdot S_{i,j+1}(t)) \quad (1)$$

where $\oplus$ and $\cdot$ are the *xor* and *and* logical operators, respectively, and c, n, s,

DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012

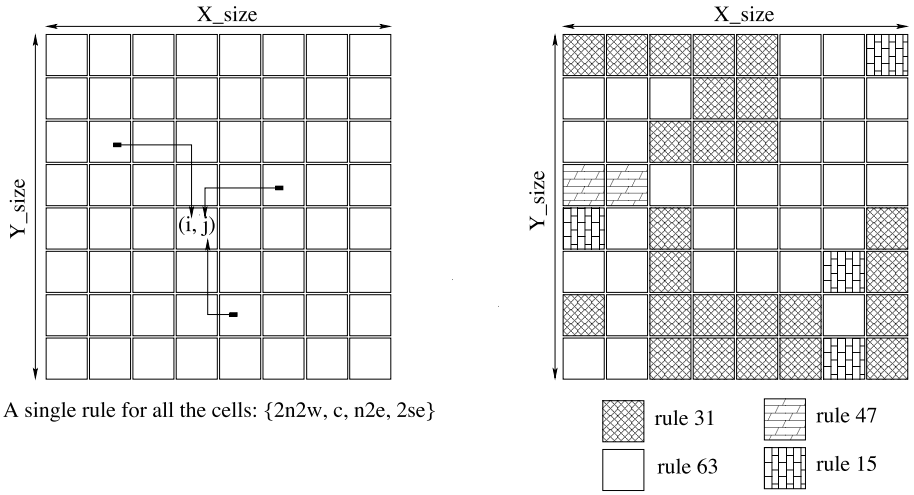


Fig. 1. Two-dimensional CA-based PRNGs: an homogeneous (left) and no-homogeneus (rigth), see [2] and [1], respectively, for details.

e, and w are binary variables, indicating whether the respective neighboring cell state is taken into account. The rule of a cell is then given by the 6-bit string $xcnwse$ and just four different transition rules were used.

Since stochastic heuristic search techniques were used to design the above cited PRNGs, there is no guarantee that the reported results are optimal and hence better 2D CAs cannot be constructed by modifying some parameters [1, 2]. Under this observation, in this paper, we evaluate the nature of 4-neighbor neighborship as a mean to improve the randomness quality. A 2D CA configuration that keeps most of the parameters proposed in [2] was used, except that different assymetric 4-neighbor neighborhoods were explored [3]. The four neighbors were restricted to be located in a 5×5 area around a center cell in order to reduce the searching space. After an exhaustive search, the neighborhood that provides the best results according to standard statistical tests, is given by $\{2sw, 2s2w, ne, 2e\}$. Unlike the CA-based PRNG proposed in [2], the center cell c, is not used in the neighborhood. According the the experimental results, section 3, the randomness quality of the found CA-based PRNG, denoted as $NPCA$ hereafter, is better than the reported in [1, 2]. Other neighborhoods were also found that provide randomness quality at least as good as those reported in the literature. Several good quality CA configurations have applications in the efficient implementation of parallel PRNGs in order to reduce correlation among numbers and autocorrelation.

3 Results and experimental evaluation

The performance of a PRNG is measured by various statistical tests, being one of the most commonly used the Diehard *battery* which consists of 18 different and independently tests. Most tests return $p-values$, which should be uniform distributed over $(0, 1)$ for truly independent random bits. Data streams, 3 million 64-bit words, were collected from the 2D CAs which were first initalized to a high-entropy state by clocking for 64 cycles from an initial

DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012

Table I. Diehard test results for three different CA-based and the Mersenne-Twister PRNGs.

Approach	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17
Sha.	0.9	1	1	0.92	0.05	0.95	0.57	0.96	1	0.84	1	1	1	1	0.90	0.75	1
Tom.	1	1	1	1	0.95	0.78	0.92	0.93	1	0.92	1	1	0.95	1	1	1	1
NPCA	1	1	1	1	1	0.95	1	0.96	1	0.96	1	1	1	1	0.91	1	1
MT	1	1	1	0.96	0.9	0.95	0.92	0.96	1	0.92	0.90	1	1	1	0.90	1	1

state of a single 1 in cell (0, 0) [2]. Results of randomness tests for different PRNGs, Shackleford (Sha.), Tomassini (Tom.), a new proposed CA-based (NPCA), and Mersenne-Twister (MT), are shown in Table I and Figure 2.

3.1 Randomness quality

The models were tested with a 95% of confidence level. Thus, when $p-value < 0.025$ (lower threshold) and $p-value > 0.975$ (upper threshold) means the PRNG fails the test, i.e, some $p-values$ are outside of the feasible area as shown in Figure 2. In Table I, p_i stands for the ratio of the $p-values$ that pass the test to the total number of $p-values$ in the $i-th$ test *(Birthday Spacing, Overlapping 5-Permutation, Binary Rank* 31×31 *and* 32×32*, Binary Rank* 6×8*, Bitstream, OPSO, OQSO, DNA, Count-The-1's 01, Count-The-1's 02, Parking Lot, Minimum Distance, 3DS Spheres, Squeeze, Overlapping Sums, Runs, Craps)*. Thus, for instance, $p_5 = 0.05$ means that Shackleford CA PRNG almost fails the *Bitstream* test since the $p-values$ are outside of the confidence region. When $p_i = 1$ ($p_i = 0$) means that all the $p-values$ fully pass (fail) the $i-th$ test.

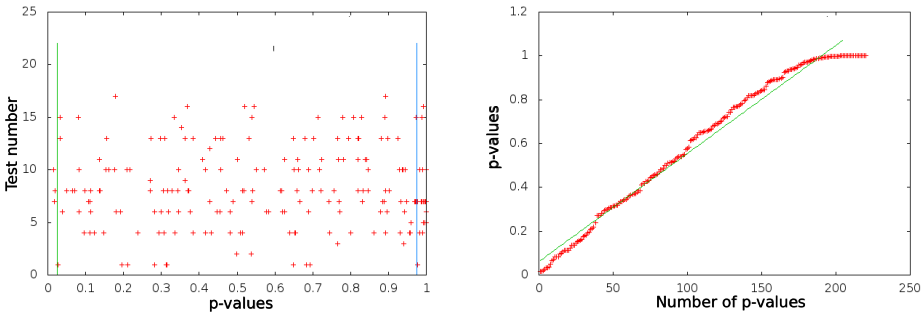
Figure 2 shows graphs for $p-value$ distributions and linear regressions for different PRNGs. The experiments show that in general the CA-based PRNGs perform well on the Diehard test suite. We analyzed the distribution of all $p-values$, which in the ideal case must fit to a rect line. The Mersenne-Twister PRNG was included in the evaluation for reference purposes as it produces among the highest-quality sequences of random bits.

From the experiments and results, it can be derived that good random-number sequences can be produced by 2D CA, however, it should be pointed out that most of them are sensitive to architectural parameters of the CA. For instance, it can be observed that if the method to extract random numbers from a given 2D CA is changed, the results migth be likely different. On the other hand, Mersenne-Twister PRNG shows to be more robust to different methods to extract and create a sequence of random numbers as tested by Diehard. Experiments on other parameter variations are carried out and for space considerations are not presented here.

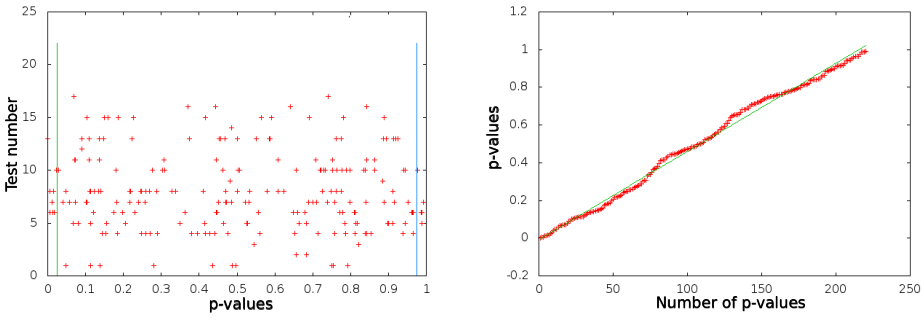
3.2 Hardware requirements

The 2D CA PRNGs were modeled using the Very High Speed Integrated Circuits Hardware Description Language (VHDL) targeted to Xilinx FPGA devices. A top-down approach was employed and the use of packages and generics were promoted in order to have a highly configurable VHDL model. The FPGA hardware resources utilization is summarized in Table II. A 3-6

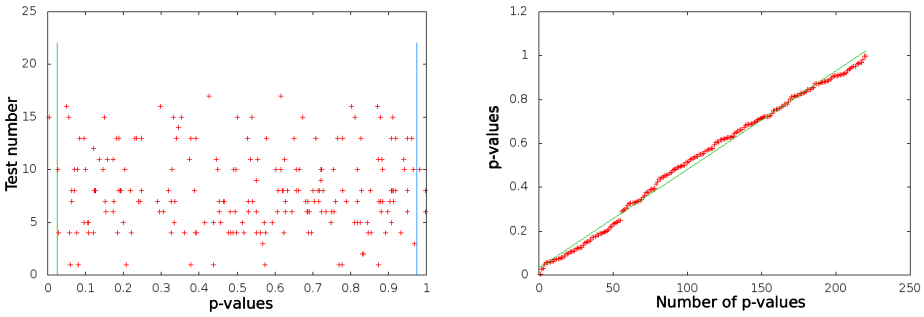
DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012



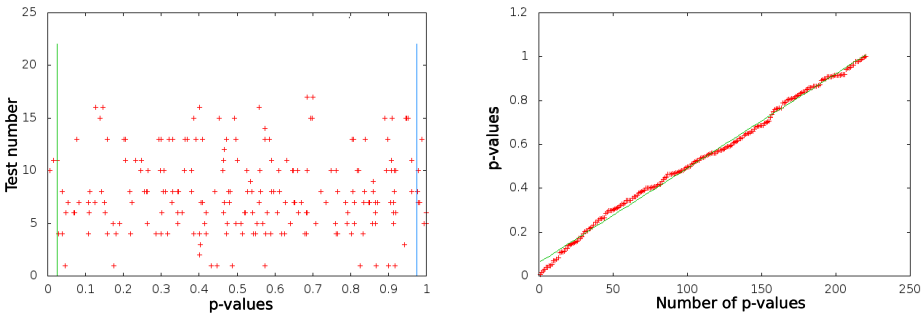
(a) Shackleford



(b) Tomassini



(c) Proposed CA



(d) Mersenne-Twister

Fig. 2. p-value distributions (left column) and linear regressions (right column) for different PRNGs.

input transition rule maps entirely into a single Virtex-6 LUT without any influence of the function complexity. Thus, the total number of LUTs/flip-flops is equal to the size of the grid used for the CA-based PRNG. The

DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012

Table II. Hardware resource utilization of CA-based PRNGs and Mersenne-Twister in a Virtex 6 XC6VLX240T device.

PRNG	LUTs	Slice Flip-Flops	Slices	Block RAM
CA-based	64	64	18	0
Mersenne-Twister	127	41	49	64×32

Xilinx place and route tools produce fairly the same results for each 2D CA-based PRNG than can be clocked at frequencies over 500 MHz. With this frequency, 500-million pseudorandom numbers per second can be generated on-chip. However, if the numbers are used off-chip, the input/output overhead must be considered and the throughput should be lower. Theoretically, it could be possible to implement around 2000 2D CA-based PRNGs on-chip that yields massive parallel pseudorandom numbers per second in the Virtex 6 device (37,680 slices). In practice, the PRNG density should be less since to capture and extract the whole bitstreams concurrently would require unrealistic databus widths. The potential massive parallel PRNGs into a single FPGA could outperform the results reported in [4], where a Graphical Processing Unit (GPU)-based was used, and in [5] where PRNGs were targeted to on-chip multiple cores. The randomness quality of CA-based PRNGs compares well with the Mersenne-Twister PRNG.

4 Conclusions

Through experiments with 2D CA, efficient means of producing PRNGs that perform well compared to well-known PRNGs have been identified. After presenting performance on standard statistical tests for randomness, we discussed tradeoffs between required hardware FPGA resource and PRNG performance. Generally, there are four aspects affecting the randomness of 2D CA-based PRNGs: boundary condition, transition rule, length of CA, and initial seed. 2D CA-based PRNGs produce good quality random numbers for potential embedded applications or for massive parallel implementations under real-time constraints due to its performance and affordable cost in silicon area. FPGAs provide a suitable substrate for CA-based PRNGs implementation due to their granularity and the underlying architectural organization match well 2D CA structures.

Acknowledgments

This work was supported in part by CONACyT, Mexico, under research grant No. 99912.

DOI: 10.1587/elex.9.1391
Received June 28, 2012
Accepted July 24, 2012
Published September 10, 2012