A NEW RANDOM NUMBER GENERATOR WITH A NEW CELLULAR AUTOMATA MODEL

Emre Göncü
Electronics and Communication Dept.
Istanbul Technical University
Turkey
egoncu@itu.edu.tr

Abdülkadir Koçdoğan
Electronics and Communication Dept.
Istanbul Technical University
Turkey
akkocdogan88@gmail.com

Müştak Yalçın
Electronics and Communication Dept.
Istanbul Technical University
Turkey
mustak.yalcin@itu.edu.tr

Abstract

Cellular Automata with Memory are Cellular Automata models in which next states of the cells depends on not only current states but also fixed previous states of the cells. In this paper, a new Cellular Automata model named Cellular Automata with Random Memory in which next states of the cells depends on randomly chosen previous states or current states of the cells is proposed. Furthermore, a Random Number Generator using Cellular Automata with Random Memory is suggested. The Cellular Automata with Random Memory and Random Number Generator is implemented in C. According to the experimental results, randomness of the random sequences generated by the Random Number Generator is greater than the randomness of the random sequences generated by the rand() function given in standard C library.

Key words

Cellular Automata, Cellular Automata with Memory, Cellular Automata with Random Memory, Random Number Generator, CARM, RNG

1 Introduction

Cellular Automata (CA) are discrete dynamical systems firstly introduced by vonNeumann and Ulam [von Neumann, 1963; Ulam, 1972]. CA is used in very different fields like modelling of physical systems [Wolf-Gladrow, 2000], random number generators [Wolfram, 1986], image processing [Rosin, 2006], *etc.* Because of their massively parallel computation and suitability for VLSI implementation, they are good candidate for Field Programmable Gate Array (FPGA) implementation and Application Specific Integration Circuit(ASIC) implementation [Kobori et al., 2001].

A lot of CA models have been proposed for various purposes since they are introduced. CA with Memory (CAM) are considered under these CA models whose next states rely on not only the current states but also the previous states of the cells, contrary to standard CA models. As far is known, one of the first CAM models has been suggested by Edward Fredkin [Toffoli and Margolus, 1990]. After his suggestion many different CAM models have been proposed. One of these models, named Elementary Cellular Automata with Memory (ECAM), has been proposed by Ramon Alonso-Sanz *et al.* [Alonso-Sanz, R., Martín, 2003]. Additionally, a different CAM model, named ECAM-T, has been proposed by Paul-Jean Letourneau [Letourneau, 2006]. Finally a new CAM model has been proposed in our previous work [Goncu and Yalcin, 2014].

In this paper, a new CA model, named CA with Random Memory (CARM) is introduced. In this model, contrary to CAM models, next states of the cells rely on random previous states instead of fixed previous states of the cells. Moreover using a CARM, a random number generator (RNG) is proposed. To verify the randomness of the random sequence generated by the RNG, FIPS 140-1[FIPS 140-1, 1994] is applied. According to the experimental results, randomness of the random sequences generated by the RNG is greater than the randomness of the random sequences generated by the rand() function given in standard C library.

2 Cellular Automata

CA are discrete dynamical systems composed of cells sited at a cellular space. Each cell has a state changing according to state of the its neighbour cells and a function called rule. Therefore CA are defined with a cellular space $\mathbb{Z}^d$, a neighbourhood V, a set of states Q and a rule f. Dynamics of the CA is given by the following

$$\alpha_i(n+1) = f\left(\alpha_{i+i_1}(n), \ldots, \alpha_{i+i_m}(n)\right) \qquad (1)$$

where $\alpha_i(n)$ denotes the state of the cell sited at $i \in \mathbb{Z}^d$ at time step n and $i_1, \ldots, i_m \in \mathbb{Z}^d$ denotes the elements of neighbourhood V. Therefore rule f is defined by $f : Q^m \to Q$.

Simplest CA, named Elementary Cellular Automata (ECA) [Wolfram, 1983] is defined with cellular space $\mathbb{Z}$, neighbourhood $V = (-1, 0, 1)$, state set $Q = \{0, 1\}$ and a rule f. The dynamics of an ECA is given by

$$\alpha_i(n+1) = f\left(\alpha_{i-1}(n), \alpha_i(n), \alpha_{i+1}(n)\right). \quad (2)$$

ECA local rules are defined with respect to all possible states of neighbor cells. In Figure 1, an example of ECA local rule is given (Black cells denote the state 1, white cells denote the state 0). Values given upper row specify the possible states of the center cells' neighbors. Considering lower row, values show the state of the center cells at the next time step. ECA local rules are named with regard to Wolfram notation[Wolfram, 1983] so that for the rule given Figure 1, rule name is decimal equivalent of the binary value (left to right) given lower row.

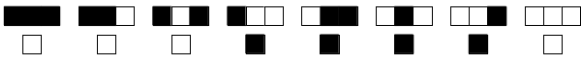

Figure 1. An example of ECA local rule: Rule 30.

Dynamic behavior of an ECA is usually analyzed by the evolution figure. Evolution of an ECA with Rule 150 contained 128 cells for 64 time steps is given in Figure 2. Considering Figure 2, first row shows the initial state of the cells, lower rows denote the states of the cells, at next time steps.

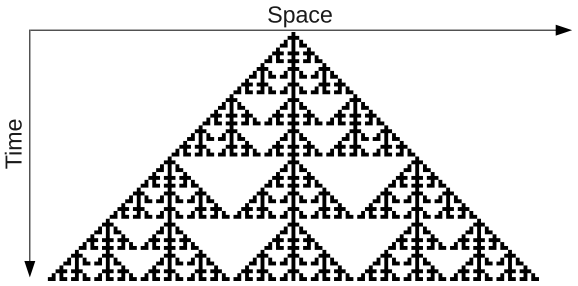

Figure 2. Evolution of an ECA with Rule 150 containing 128 cells for 64 time steps.

3 Cellular Automata with Memory

In the concept of CAM, next states rely on not only the current states but also the previous states of the cells, contrary to standard CA models.

As far is known, one of the first CAM models has been suggested by Edward Fredkin [Toffoli and Margolus, 1990]. Dynamics of the model has been given by

$$\alpha_i(n+1) = f(\alpha_{i-1}(n), \alpha_i(n), \alpha_{i+1}(n), \alpha_i(n-1)). \quad (3)$$

After ten years later from Fredkin's proposal, Ramon Alonso-Sanz *et al.* have propounded a new model for CAM named Elementary Cellular Automata with Memory (ECAM) [Alonso-Sanz, R., Martín, 2003]. Dynamics of ECAM is defined by two functions, majority function and standard local transition function which is

$$\alpha_i(n+1) = f(m_{i-1}(n), m_i(n), m_{i+1}(n)) \quad (4)$$

where f is one of the ECA local rules and m_i is the majority function that results in most frequent state for the all past states of cell i. Later, Alonso-Sanz *et al.* have changed the ECAM model with some constraints[Alonso-sanz and Martín, 2005], namely the majority function is defined with considering instead of all past states, only last three states of the cells. Hence the $m_i(t)$ in (4) can be given by

$$m_i(n) = s(\alpha_i(n), \alpha_i(n-1), \alpha_i(n-2)) \quad (5)$$

where s denotes the majority function.

In Figure 3, evolution of an ECAM contained 128 cells for 64 time steps is given. The ECAM is defined by standard local transition function Rule 150 and majority function considering last three states.

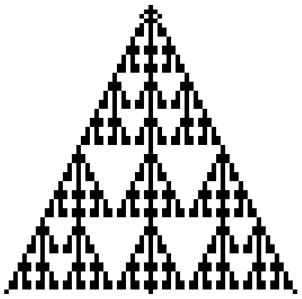

Figure 3. Evolution of an ECAM (considering last three states) containing 128 cells with Rule 150 for 64 time steps.

Additionally, a different CAM model, named ECAM-T, has been proposed by Paul-Jean Letourneau [Letourneau, 2006]. Dynamics of the ECAM-T model can be given by

$$\alpha_i(n+1) = f(\alpha_{i-1}(n), \alpha_i(n-T), \alpha_{i+1}(n)). \quad (6)$$

Considering (6), the memory is mentioned only for center cell (cell i). Evolution of an ECAM-T ($T = 2$) containing 128 cells for 64 time steps with same initial states in the manner of the ECAM and the ECA given above is given in Figure 4. Local transition function of considered ECAM-2 is specified as Rule 150.

Finally, a different CAM model has been proposed in our previous work [Goncu and Yalcin, 2014]. Dynamics of a proposed CAM is given by

$$\alpha_i(n+1) = f(\alpha_{i-1}(n-x), \alpha_i(n-y), \alpha_{i+1}(n-z)) \quad (7)$$

where x, y and z are non-negative integers which determine the corresponding past states of the neighbour

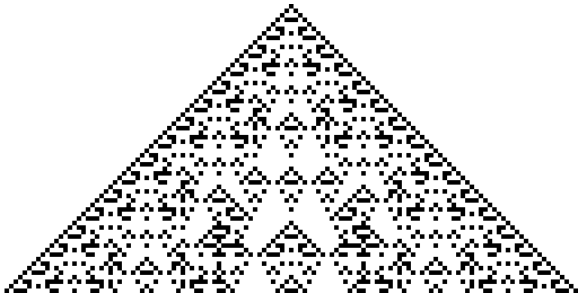

Figure 4. Evolution of an ECAM-2 with Rule 150 contained 128 cells for 64 time steps.

cells. Figure 5 illustrates the evolution of the new CAM containing 128 cells for 64 time steps with same initial states in the manner of CAMs given above for Rule 30 and $x = 1$,$y = 1$ and $z = 2$.

Figure 5. Evolution of the new CAM contained 128 cells with Rule 30 for $x = 1, y = 1$ and $z = 2$ and for 64 time steps.

4 Cellullar Autama with Random Memory

In this paper a new CA model, named CARM is proposed. In this model, contrary to CAM models, next states of the cells rely on randomly chosen previous states instead of fixed previous states of the cells. Dynamics of the CARM is given as the following

$$\alpha_i(n+1) = f\Big(\alpha_{i+i_1}(n-\tau_1(n)), \ldots, \alpha_{i+i_m}(n-\tau_m(n))\Big) \quad (8)$$

where $\tau_1(n), \ldots, \tau_m(n) < n$ are random non-negative integers.

A special CARM can be defined with cellular space $\mathbb{Z}$, vector $V = (-1, 0, 1)$, state set $Q = \{0, 1\}$ and a rule f. Dynamics of this CARM is given by

$$\alpha_i(n+1) = f\Big(\alpha_{i-1}(n-\tau_1(n)),\alpha_i(n-\tau_2(n)), \alpha_{i+1}(n-\tau_3(n))\Big) \quad (9)$$

where $\tau_1(n), \tau_2(n), \tau_3(n) \in \{0, 1\}$ are random integers. Evolution of the special CARM contained 128 cells with Rule 150 for 64 time steps is given in Figure 6.

Figure 6. Evolution of the special CARM contained 128 cells with Rule 150 for 64 time steps.

4.1 Random Number Generation using the CARM

In this paper a novel RNG using the special CARM, is proposed. Random bits are generated by Exclusive-OR (XOR) operation on states of the two specified cells. Therefore for every time steps, a random bit is generated during the evolution of the CARM. Block diagram of the RNG is given in Figure 7, where $r(n)$ denotes the random bit at time step n.

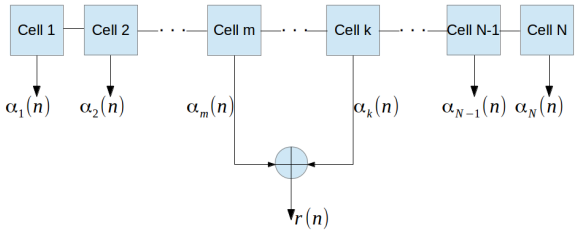

Figure 7. Block diagram of the RNG.

5 Experimental Results

In this paper the proposed RNG is constituted by the special CARM composed of 256 cells with Rule 150. States of the 25th and 127th cells are chosen for XOR operation. To verify the randomness of the random sequence generated by the RNG, FIPS 140-1[FIPS 140-1, 1994] is applied. The RNG and the FIPS 140-1 tests are implemented in C and run on a Windows 7 platform. Moreover randomness of the random sequences $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$ is analysed by the FIPS 140-1, as well. Notice that all random sequences generated from time steps $n = 256$ to $n = 20255$.

Table 1 illustrate the test results for the random sequence generated by the proposed RNG and the random sequences $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$. According to the results, random sequence generated by proposed RNG passes all tests of FIPS 140-1 while $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$ fails. Therefore, proposed RNG increases randomness of the random sequence generated by rand() function in standard C library.

Table 1. Test results for FIPS 140-1

	Required Values	$r(n)$	$\tau_1(n)$	$\tau_2(n)$	$\tau_3(n)$
Monobit Test	9654-10346	9978	10000	10001	10000
Runs Test('1's) Length=1	2267-2773	2534	0	4765	0
Runs Test('1's) Length=2	1079-1421	1256	0	2618	1
Runs Test('1's) Length=3	502-748	639	0	0	0
Runs Test('1's) Length=4	223-402	304	2305	0	742
Runs Test('1's) Length=5	90-223	152	156	0	1406
Runs Test('0's) Length=1	2267-2773	2549	0	4766	1
Runs Test('1's) Length≥ 6	90-223	165	0	0	0
Runs Test('0's) Length=2	1079-1421	1242	0	2616	0
Runs Test('0's) Length=3	502-748	616	0	0	0
Runs Test('0's) Length=4	223-402	319	2304	0	742
Runs Test('0's) Length=5	90-223	142	156	0	1406
Runs Test('0's) Length≥ 6	90-223	165	0	0	0
Poker Test	1.03-57.40	11.2192	5000.08	5118.84	5639.23
Long Runs('0's)	> 34	None	None	None	None
Long Runs('1's)	> 34	None	None	None	None
Pass/Fail	-	Pass	Fail	Fail	Fail

6 Conclusion

In this paper, a new CA model, named CARM, is proposed. Moreover using CARM, a new RNG is suggested. Random sequence generated by the RNG passes all FIPS 140-1 tests while random sequences $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$ does not. This result shows that the CARM increases the randomness of the random sequences $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$. Because the function f defining the dynamics of the CARM depends on $\tau_1(n)$, $\tau_2(n)$ and $\tau_3(n)$ (see Equation (9)).

As a future work, the CARM will be designed with a digital circuit containing delay lines for the randomly chosen memories. Therefore a true RNG will be designed with the CARM.

References

Alonso-sanz, R. and Martín, M. (2005). One-dimensional Cellular Automata with Memory in Cells of the Most Frequent Recent Value. 15:203–236.

Alonso-Sanz, R., Martín, M. (2003). Elementary cellular automata with memory. *Complex Systems*, 14:99–126.

FIPS 140-1 (1994). Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-1. *U.S. Department of Commerce/NIST, National Technical Information Service*. Springfield, VA.

Goncu, E. and Yalcin, M. (2014). A new cellular automata model with memory and its fpga implementation. In *Cellular Nanoscale Networks and their Applications (CNNA), 2014 14th International Workshop on*, pages 1–2.

Kobori, T., Maruyama, T., and Hoshino, T. (2001). A cellular automata system with FPGA. *9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM01)*.

Letourneau, P. (2006). *Statistical Mechanics of Cellular Automata with Memory*. Canadian theses. University of Calgary (Canada).

Rosin, P. L. (2006). Training cellular automata for image processing. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 15(7):2076–87.

Toffoli, T. and Margolus, N. H. (1990). Invertible cellular automata: A review. *Physica D: Nonlinear Phenomena*, 45(13):229 – 253.

Ulam, S. M. (1972). Some ideas and prospects in biomathematics. *Annual Review of Biophysics and Bioengineering*, pages 277–292.

von Neumann, J. (1963). The general and logical theory of automata. *Collected Works*, 5:288.

Wolf-Gladrow, D. (2000). *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction*, volume 308. Springer, Berlin.

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55:601–644.

Wolfram, S. (1986). Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 7(2):123–169.