VLSI DESIGN
1998, Vol. 7, No. 2, pp. 177–189
Reprints available directly from the publisher
Photocopying permitted by license only

Non-linear Hybrid Cellular Automata as Pseudorandom Pattern Generators for VLSI Systems

I. KARAFYLLIDIS*, I. ANDREADIS, PH. TSALIDES and A. THANAILAKIS

Department of Electrical and Computer Engineering, Democritus University of Thrace, GR-671 00 Xanthi, Greece

(Received 1 December 1995)

The concept of hybrid in space-time Cellular Automata is introduced, for the first time, in this paper, and it is suggested that non-linear hybrid in space-time Cellular Automata can be used as pseudorandom pattern generators for VLSI systems, because they can produce patterns with various densities of "1", distributed at will in space and time. The cycle lengths of non-linear hybrid Cellular Automata can be estimated using Lyapunov exponents.

Keywords: Pseudorandom pattern generation, testing, VLSI, cellular automata

1. INTRODUCTION

Pseudorandom pattern generators are mainly used for the generation of input test vectors for a VLSI system under test. They are also used in parallel processing by fine grained VLSI arrays [1]. Pseudorandom Number Generators (PRNG's) based on Linear Cellular Automata have been found to generate more random patterns than those generated by Linear Feedback Shift Registers (LFSR's). Cellular Automata compare very favourably with the auto correlation and cross correlation of LFSR generators. Extensive comparisons between Cellular Automata and LFSR's can be found in [1, 2]. Cellular Automata do not have long feedback loops, and they operate at higher speeds. Their regular structure and the local interconnections result in efficient layouts and less silicon overhead [3]. Linear Cellular Automata as exhaustive and pseudoexhaustive pattern generators and signature analysers for Built-In Self-Test, have been investigated by several authors [4–7]. It has been shown that only by the rules 90 and 150, linear Cellular Automata can generate exhaustive and pseudoexhaustive test patterns. Linear Cellular Automata have been extensively investigated, but a little effort has gone into the non-linear Cellular Automata, although they generate more random patterns and have more complex behaviour than the linear ones [8]. This is due to the fact that the evolution of non-linear Cellular Automata cannot be described using finite mathematical formulas, but using computer simulations and statistical methods. To the best of our

*Corresponding author.

knowledge non-linear hybrid Cellular Automata have never been investigated.

The aim of this work is to introduce the concept of hybrid in space-time Cellular Automata as PRNG's for VLSI systems, to show that these Cellular Automata can be used as pseudorandom pattern generators for VLSI systems, because they can produce patterns with various densities of "1" distributed at will in space and time, to introduce Lyapunov exponent as a measure of pattern randomness and to show that given a non-linear hybrid in space-time Cellular Automaton, desirable cycle lengths can be achieved by proper choice of the initial global state and the time period in which the second rule is activated. In Section 2 the necessary background on the Cellular Automata theory is presented. Section 3 shows that Cellular Automata can not generate random but pseudorandom patterns. In Section 4 the generation of pseudorandom patterns by non-linear hybrid Cellular Automata is investigated.

2. CELLULAR AUTOMATA

A Cellular Automaton is an artificial universe in which space and time are discrete. A cellular automaton consists of a n-dimensional regular uniform lattice (array), which may be infinite in extent. At each site of the lattice (cell), a physical quantity takes on values, which are elements of a finite commutative ring $\mathbf{R}_k$ with k elements. This physical quantity is the global state of the Cellular Automaton, and the value of this quantity at each particular cell is the state of this cell. The states of the cells are updated simultaneously at discrete time steps. Furthermore, the state of a cell is affected only by the states of the cells in its neighbourhood at the previous time step.

A Cellular Automaton can be characterised by five properties:

1) The number of spatial dimensions (n).
2) The width of each side of the array (w). w_j is the width of the j-th side of the array, where $j=1,2,3,...,n$.
3) The width of the neighbourhood of the cell (d). d_j is the width of the neighbourhood at the j-th side of the array.
4) The number k of the elements of the commutative ring $\mathbf{R}_k$.
5) The Cellular Automaton rule, which is an arbitrary function F over $\mathbf{R}_k$. The state of a cell at time step $(t+1)$ is computed according to F. The dependent variable of F is the state of the cell at time $(t+1)$, and the independent variables are the states of the cells in its neighbourhood at the time step (t). The function F may not be the same for all the cells of the Cellular Automaton. If more than one functions are used, then the Cellular Automaton is hybrid. However, only one function may be used for each cell.

Cellular Automata have been extensively used as computational models for physical systems simulation [9–12], leading to algorithms which are appropriate for an implementation on a massively parallel computer, such as CAM-8 [13].

One dimensional Cellular Automata with $d=3$ and $\mathbf{R}_k=Z_2$ are called elementary Cellular Automata. Elementary Cellular Automata have been extensively studied by Wolfram [14, 15]. An elementary Cellular Automaton evolves according to 256 possible distinct rules. Those rules are numbered by Wolfram from 0 to 255. Some of those rules are shown in the table of Figure 1. The first row of the table contains the possible states of a cell and its two neighbours (right and left) at

t	111	110	101	100	011	010	001	000	Wolfram law
t+1	0	0	0	0	0	0	0	1	1
	0	0	0	0	0	0	1	0	2
	0	0	0	0	0	0	1	1	3
	0	0	0	1	1	1	1	0	30
	0	1	0	1	1	0	1	0	90
	1	0	1	1	0	1	0	0	180
	1	1	1	1	1	1	1	1	255

FIGURE 1 A list of some of Wolfram rules. The states of the cell at time $t+1$ are converted into the rule number. For example $10010110_2=150_{10}$.

time t. All the other rows contain the state of the cell at time $t+1$, according to the rule which is contained in the last column of the table. The second row contains the state of the cell at time $t+1$, according to rule 1, the third row contains the state of the cell at time $t+1$, according to rule 2 and so on. At the ends of an one-dimensional finite array of cells two boundary conditions can be applied. The null boundary condition, according to which the first cell is receiving a constant 0 input from its left and the last cell a constant 0 input from its right, and the periodic boundary condition, according to which the left neighbour of the first cell is the last cell and the right neighbour of the last cell is the first cell.

Elementary Cellular Automata are of simple construction but they can exhibit complex and varied behaviour. Moreover, due to the local interconnections and the discreteness in space and time, synchronous VLSI systems can be used as an implementation medium. In Figure 2 the evolution of a Cellular Automaton which evolves according to rule 90 is shown. The rule 90 is the addition of the left and right neighbour of the cell performed over Z_2 (logical XOR operation)

$$c_j^{t+1} = c_{j-1}^t \oplus c_{j+1}^t \tag{1}$$

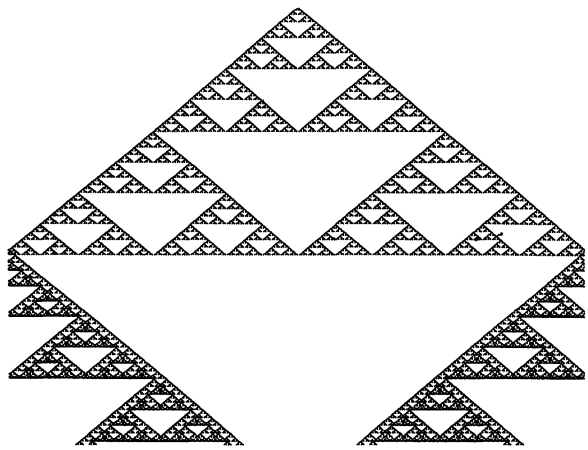


FIGURE 2 A Cellular Automaton evolving in time from a single non-zero initial state according to rule 90.

where c_j^{t+1} is the state of j-th cell at time $t+1$ and c_{j-1}^t, c_{j+1}^t are the states of the $j-1$ and $j+1$ cells at time t.

The Cellular Automaton in Figure 2 consists of 501 cells and the evolution for 450 time steps is shown. Periodic boundary conditions are imposed. The "1" is represented by a black dot and the absence of a black dot represents the "0". The initial global state contains only one non zero cell state, the state of cell 251. The evolution pattern is self similar and strongly resembles to the Sierpinski Gasket [16]. The pattern is characterised by a fractal dimension 1,585.

In the rest of this work only the elementary one-dimensional Cellular Automata will be studied, and the term "Cellular Automaton" will be used instead of the term "Elementary one-dimensional Cellular Automaton".

For one-dimensional Cellular Automata the rule is given by the following relation:

$$c_j^{t+1} = F(c_{j-1}^t, c_j^t, c_{j+1}^t) \tag{2}$$

If the function F is, or can be reduced to, a linear form, then the Cellular Automaton law is linear. A Cellular Automaton evolving according to a linear law is a Linear Cellular Automaton. There are eight possible linear laws:

$$\begin{aligned}
&\text{Law } 0 \quad c_j^{t+1} = 0\\
&\text{Law } 60 \quad c_j^{t+1} = c_{j-1}^t \oplus c_j^t\\
&\text{Law } 90 \quad c_j^{t+1} = c_{j-1}^t \oplus c_{j+1}^t\\
&\text{Law } 102 \quad c_j^{t+1} = c_j^t \oplus c_{j+1}^t\\
&\text{Law } 150 \quad c_j^{t+1} = c_{j-1}^t \oplus c_j^t \oplus c_{j+1}^t\\
&\text{Law } 170 \quad c_j^{t+1} = c_{j+1}^t\\
&\text{Law } 204 \quad c_j^{t+1} = c_j^t\\
&\text{Law } 240 \quad c_j^{t+1} = c_{j-1}^t,
\end{aligned} \tag{3}$$

where $\oplus$ represents the addition over Z_2 (logical XOR operation). All these rules, other than those in (3), are non-linear. Linear Cellular Automata have been extensively studied [17–19], and can

easily be implemented by VLSI systems. Each cell consists of one flip-flop, which holds the cell state, and a few gates for the implementation of the rule. Three cells of a Cellular Automaton, which evolve according to rule 90, are shown in Figure 3.

3. PSEUDORANDOM PATTERN GENERATION BY CELLULAR AUTOMATA

Cellular Automata as pseudorandom pattern generators for Built-InSelf-Test have been proposed by Hortensius *et al.* [1, 2]. They examined a variety of Cellular Automata rules using computer simulations. Patterns generated by Cellular Automata were tested using the Knuth [20] random number tests. These tests are used as a metric against which a pseudorandom pattern generator can be measured for randomness. The Knuth random tests are [20]:

1. Equidistribution: numbers should be uniformly distributed over the input space.
2. Serial: successive pairs, triples, etc., should be uniformly distributed.
3. Gap: length of gaps between numbers should be binomially distributed.
4. Poker: the number of k tuples with r different values should be correctly distributed.
5. Coupon Collection: the number of sequence values required to obtain a complete collection of numbers should be correctly distributed.
6. Permutation: in blocks of length q, the $q!$ possible orderings should be equidistributed.

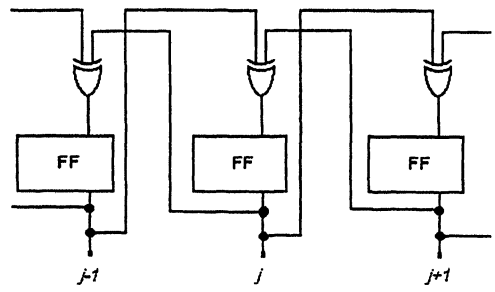


FIGURE 3 Three cells of a rule 90 Cellular Automaton.

7. Run: the length of monotonically increasing blocks in the sequence should be correctly distributed.
8. Maximum of t: the maximum values in blocks of length t should have a power law distribution.
9. Correlation: the correlation among numbers and auto correlation and cross correlation between bits of the numbers should be close to zero.

The correlation values between states of immediately neighbouring Cellular Automaton cells were found to be high. However, this correlation was found to die out 3–4 sites away. This phenomenon leads to the definition of the site spacing parameter Y ($Y=1,2,3,...$) [2]. The "random" pattern must be formed by taking the states of cells Y sites apart from each other. For example if $Y=2$, the pattern must be formed by the states of the $1,4,7,10,...$ cells. The generated patterns pass the Knuth tests for values of $Y \geq 3$. The explanation for this phenomenon will be given below.

The global state of a Cellular Automaton with n cells, at a time step t, may be represented by:

$$G = \{c_1, c_2, c_3, ...c_n\}, \quad \text{where} \quad c_j \in Z_2, j = 1,2,...n \tag{4}$$

This representation of the global state will be named "mod 2" representation. In the same way the "mod 8" representation of the global state of the Cellular Automaton, at a time step t, is given by:

$$G(\text{mod } 8) = \{e_1, e_2, e_3, ...e_n\}, \quad \text{where} \quad e_j \in Z_8, \quad j = 1,2,...n \tag{5}$$

The state of the j-th cell, e_j, in mod 8 representation, is given by:

$$e_j = c_{j-1}2^2 + c_j 2 + c_{j+1} \tag{6}$$

As an example a Cellular Automaton with 8 cells is shown in Figure 4 in mod 2 and mod 8 representa-

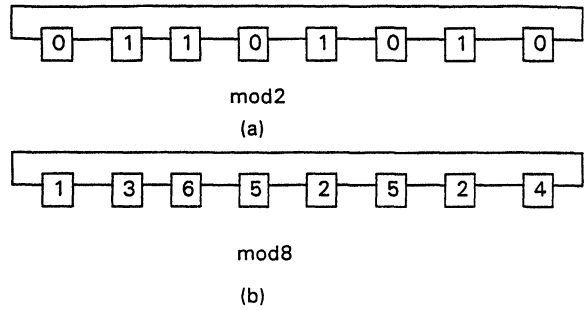


FIGURE 4 The states of an 8 cell Cellular Automaton: (a) in mod 2 representation, and (b) in mod 8 representation.

tions. The state of the third cell is 1 in mod 2 representation. The states of the second and fourth cell are 1 and 0, respectively. The state of the third cell in mod 8 representation is 6 ($6_8 = 110_2$). According to (6), the state of the j-th cell in mod8 representation is given by:

$$e_j = 2e_{j-1} + b_{j+1} \tag{7}$$

where e_{j-1} is the state of the cell $j-1$ in mod 8 representation and b_{j+1} is the state of cell $j+1$ in mod 2 representation. All states are refered to the same time step t. In mod 8 representation, according to (7), given the state of cell $j-1$, the j-th cell can assume only two possible states, because b_{j+1} can be either 1 or 0. For example, if the state of cell $j-1$ is 5, then the state of the j-th cell can only be 2 or 3. In the table of Figure 5, the possible states of the j-th cell for given states of cell $j-1$ are shown. In the table of Figure 6 the j-th cell of a Cellular Automaton is at state 5. Cell $j+1$ can assume the states 2 or 3, while cell $j+2$ can assume only the states 4, 5, 6 and 7. Cell $j+3$ can assume any of the 8 possible states. Thus the state of cell $j+3$ is completely uncorrelated with the state of the j-th cell. This fully explains the phenomenon described in [1] and [2] where the generated patterns pass the Knuth tests for values of $Y \geq 3$.

The state of the j-th cell at time step t is determined by the states of cells $j-1$, j and $j+1$ at time step $t-1$, by the states of cells $j-2$, $j-1$, j, $j+1$ and $j+2$ at time step $t-2$ and so on. The state

e_{j-1}	e_j
0	0
	1
1	2
	3
2	4
	5
3	6
	7
4	0
	1
5	2
	3
6	4
	5
7	6
	7

FIGURE 5 The possible states of the j-th cell for given states of the $j-1$ cell.

e_i	e_{i+1}	e_{i+2}	e_{i+3}
5	2	4	0
			1
		5	2
		6	3
			4
			5
	3	7	6
			7

FIGURE 6 Possible states of the cells e_{i+1}, e_{i+2} and e_{i+3}, when the state of the cell e_i is 5.

of the j-th cell at time step t affects the states of cells $j-1, j$ and $j+1$ at time step $t+1$, the states of cells $j-2, j-1, j, j+1$ and $j+2$ at time step $t+2$ and so on. Therefore, the state of the j-th cell at time step t is correlated with the values of the neighbouring cells at pervious and future time steps. Figure 7 shows the "correlation cones" of the j-th cell at time step t in Cellular Automaton space-time. It is evident that correlation, both in space and time, dies out a few sites away to the right or to the left of the j-th cell, and a few time steps before and after time step t.

It has been shown that Cellular Automata cannot generate random patterns because of the correlation between states of neighbouring cells in space and time. The states of the cells at various time steps are, in principle, predictable and, consequently, patterns generated by Cellular Automata are pseudorandom [21]. Linear Cellular Automata cannot generate satisfying pseudorandom patterns, because their evolution can be fully determined if their initial state is known [8]. Patterns, generated by Linear Cellular Automata with arbitrary initial states, can be obtained by the appropriate superposition of patterns produced by a single non-zero initial state [8, 22].

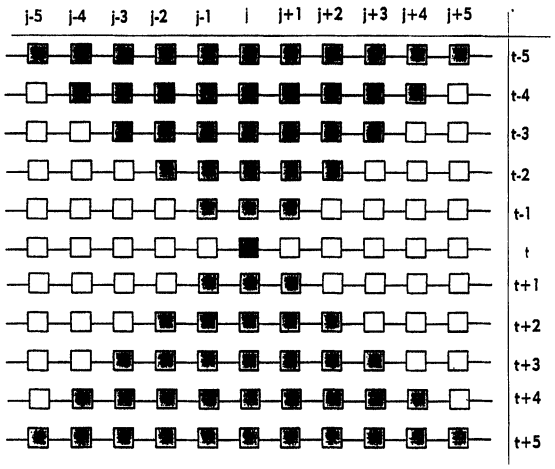


FIGURE 7 The "correlation cones" of the j-th cell at time step t in Cellular Automaton space-time.

4. PSEUDORANDOM PATTERN GENERATION BY NON-LINEAR HYBRID CELLULAR AUTOMATA

The evolution of non-linear Cellular Automata cannot be described using finite mathematical formulas, and a complete analysis of these systems is fundamentally impossible [17]. The only means for the description of non-linear Cellular Automata evolution are algorithms and extensive computer simulations of the evolution patterns, the use of statistical quantities which characterise the Cellular Automaton evolution, and the use of Lyapunov exponents which characterise the stability and the behaviour of dynamical systems [15, 23].

For Cellular Automata, which are going to be used as pseudorandom pattern generators, one of the most important statistical quantities is $P(1)$, the average density of cells with state "1" [14]. This quantity is associated with the probability for a cell to assume a "0" or an "1" state. Unlike the Linear Feedback Shift Registers, in which the probability for a site to be "1" or "0" is 1/2, this probability is different for the various Cellular Automaton rules.

For the rule 18, according to which:

t	111	110	101	100	011	010	001	000
$t+1$	0	0	0	1	0	0	1	0

where only in two out of eight cases the next state is "1", the average density of "1" is equal to 0.23. For the rule 182, according to which:

t	111	110	101	100	011	010	001	000
$t+1$	1	0	1	1	0	1	1	0

where in five out of eight cases the next state is "1", the average density of "1" is equal to 0.74.

The weighting techniques used for LFSR's are not necessary in the case of Cellular Automata. Instead, the Cellular Automaton rule which produces patterns with density $P(1)$ equal, or

almost equal, to the desirable density should be chosen. Moreover, the density $P(1)$ in a Cellular Automaton pattern does not depend on the initial state of the Cellular Automaton.

In Figure 8 the density $P(1)$ of the Cellular Automaton rule 18 for 501 cells and 450 time steps is shown. In Figures 8a and 8b the density $P(1)$ in the initial state is 20%, and 50%, respectively. In both cases, after a few time steps the average density of the rule is reached. In Figure 8c, the density $P(1)$ in the initial state is 80%. This density is quite far away from the average density of the rule. During the first few time steps, the density $P(1)$ drops below the average density of the rule. But after 50 time steps, $P(1)$ reaches the density of the rule. In both Figures 8b and 8c, the line describing the drop of $P(1)$ from 50% and 80%, respectively, is very close and almost parallel to the $P(1)$ axis. The Cellular Automaton started its evolution with an initial state with $P(1)$ about four times the density of the rule and reached this density after 50 time steps.

A Cellular Automaton may be hybrid in space or hybrid in space-time. A Cellular Automaton is hybrid in space if more than one rules are applied to its cells, for all time steps. A block diagram representing a hybrid in space Cellular Automaton

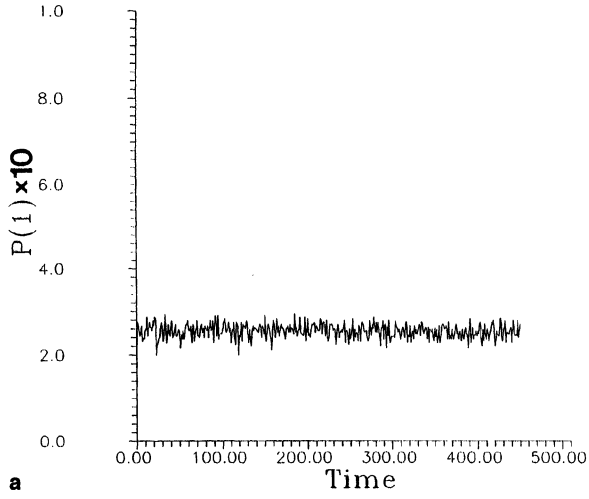
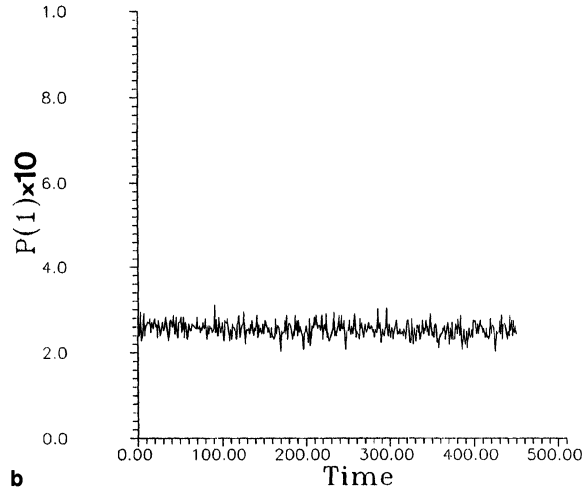


FIGURE 8 The density $P(1)$ of a rule 18 Cellular Automaton evolving from an initial state with: $P(1) = 0.2$, b). $P(1) = 0.5$ and c. $P(1) = 0.8$.



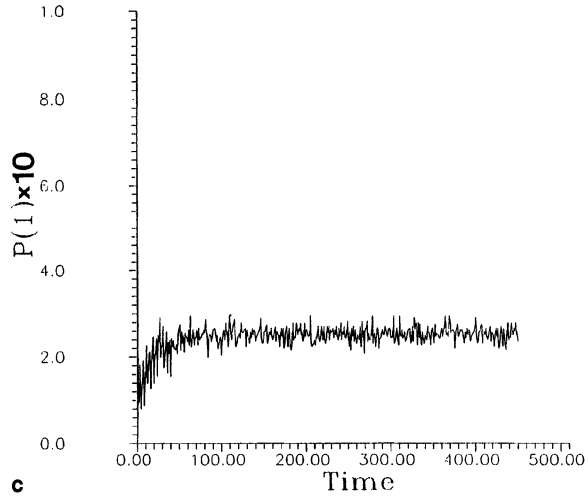


FIGURE 8 (Continued).

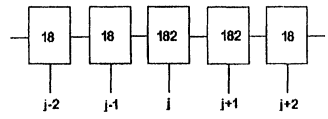


FIGURE 9 A block diagram representing a hybrid in space Cellular Automaton.

is shown in Figure 9. Two rules are applied to the Cellular Automaton cells: rule 18 and rule 182. Figure 10 shows the evolution of a hybrid in space

Cellular Automaton, with 501 cells and for 450 time steps. From cell 1 to cell 99 and cell 270 to cell 501, rule 18 is applied. On cell 100 to cell 269 the law 182 is applied.

A Cellular Automaton is hybrid in space-time if more than one rules are applied to its cells, but not for all time steps. A block diagram representing a hybrid in space-time Cellular Automaton, is shown in Figure 11. The same two rules are applied as before, on the Cellular Automaton cells. When the signal cng in Figure 11 is activated, the output of the cell with the rule 182 is transferred to the Cellular Automaton output j, whereas, if the signal cng is inactive the output of the cell with the rule 18 is transferred to the Cellular Automaton output j. Figure 12 shows the evolution of a hybrid in space-time Cellular Automaton , with 501 cells for 450 time steps. Rule 18 is applied for all time steps on cell 1 to cell 99 and also on cell 270 to cell 501. The rule 182 is applied on cell 100 to cell 269, only for time step 80 to time step 290, the rule 18 being applied on the same cells for all other time steps. Figure 13 shows the density $P(1)$ for the hybrid in space Cellular Automaton of Figure 10. The average value of $P(1)$ is 41%. It is reminded that $P(1)$ for the rules 18 and 182 is 23% and 75%,

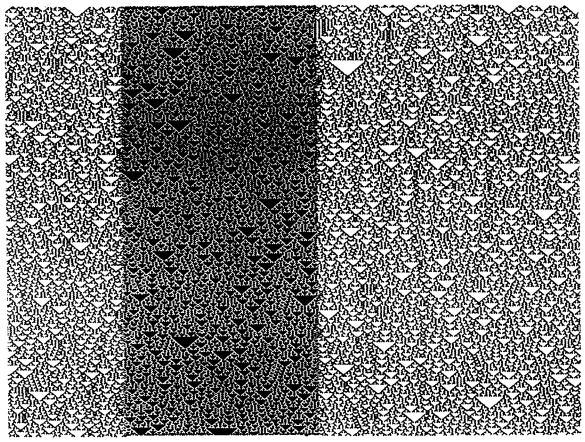


FIGURE 10 The evolution of a hybrid in space Cellular Automaton.

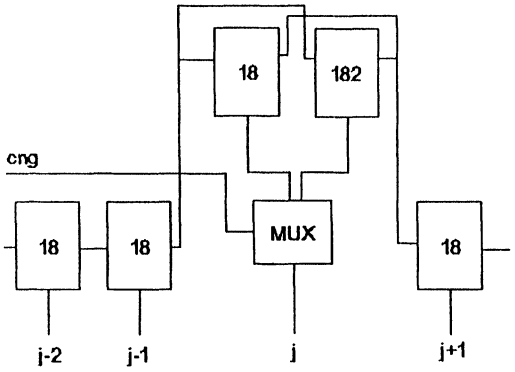


FIGURE 11 A block diagram representing a Cellular Automaton hybrid in space-time.

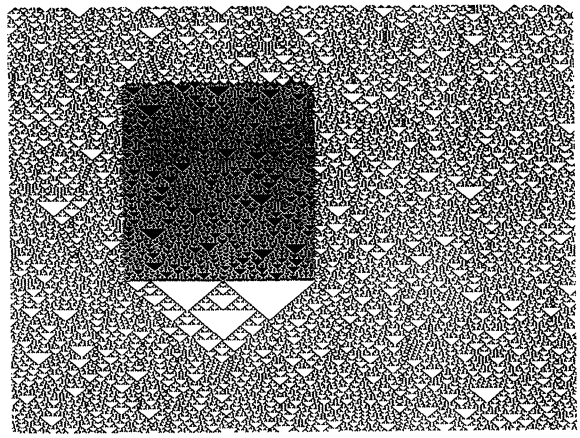


FIGURE 12 The evolution of a hybrid in space-time Cellular Automaton with 501 cells and for 450 time steps.

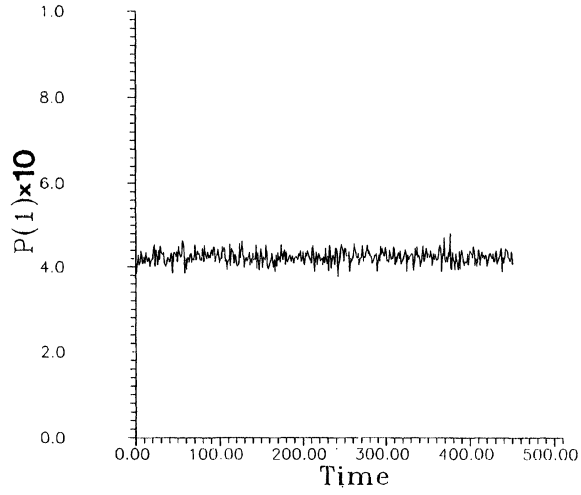


FIGURE 13 The density $P(1)$ for the Cellular Automaton of Figure 10.

respectively. Figure 14 shows the density $P(1)$ for the hybrid in space-time Cellular Automaton of Figure 12. It is evident that the average value of $P(1)$ is increased from 23% to 41%, when rule 182 is activated. Hybrid in space-time Cellular Automata can produce patterns with various densities $P(1)$, distributed at will in space and time.

Finite Cellular Automata, when evolving, enter in cycles. The cycle length of an appropriate pseudorandom pattern generator should be large. Unlike the linear Cellular Automata, the cycle length of non-linear hybrid Cellular Automata cannot be estimated using mathematical formulas. It can only be estimated with extensive computer simulations. The 256 Cellular Automata rules may be combined in two or more to form hybrid Cellular Automaton rules. Moreover, these rules may be active at various evolution time steps. The number of possible non linear hybrid Cellular Automaton rules is extremely large. This requires the search of non linear hybrid Cellular Automata with large cycle lengths, a very difficult and time-consuming task. The search space can be effectively narrowed using the Lyapunov exponents.

Lyapunov exponent λ is a measure of the sensitivity of an interator on its initial conditions [23, 25]. It quantifies the average growth of infinitesimally small errors induced in the initial condition of an interator. Generally the Lyapunov exponent is given by [25]:

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \ln|f(x_{k-1})|, \tag{8}$$

where $f(x_{k-1})$ can be substituted by:

$$\frac{1}{n} \sum_{k=1}^{n} \ln|f(x_{k-1})| = \lim_{E_0\to 0} \frac{1}{n} \sum_{k=1}^{n} \ln\left|\frac{E_k}{E_{k-1}}\right|. \tag{9}$$

E_{k-1} and E_k are the errors in $k-1$ and k interations, respectively.

It has been shown that the Lyapunov exponents can be defined for Cellular Automata, using the Boolean derivative [23, 24]. In [23] the finite-time Lyapunov exponent is defined as:

$$\lambda = \frac{1}{T} \sum_{t=1}^{T} \log n(t) \tag{10}$$

$n(t)$ is the local expansion rate of defects and is given by:

$$n(t) = \left|\frac{\sum_i N_i(t+1)}{\sum_i N_i(t)}\right| \tag{11}$$

where $N_i(0) = 1$ and

$$N_i(t+1) = \sum_j F'_{i,j}\, N_j(t) \tag{12}$$

$F'_{i,j}$ is the Boolean derivative of the Cellular Automaton rule and is given by:

$$F'_{i,j} = \frac{\partial c_i^{t+1}}{\partial c_j^t} \tag{13}$$

Consider that two Cellular Automata with the same number of cells, the same boundary conditions, the same rule and the same initial state, are

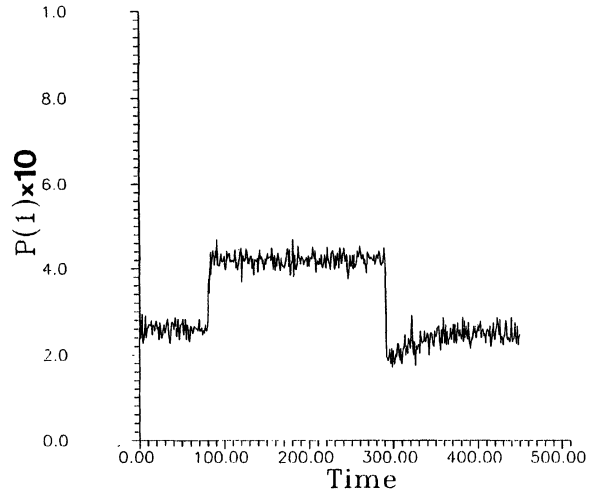
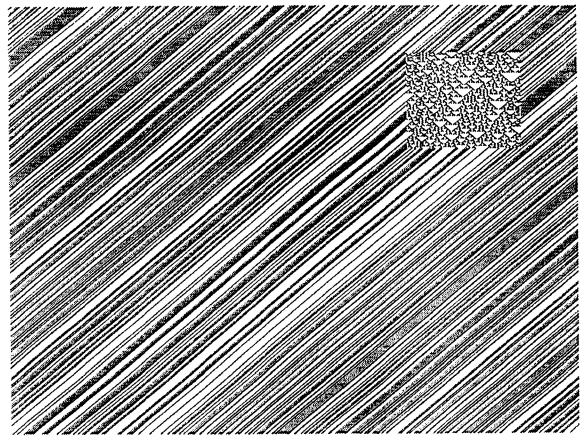


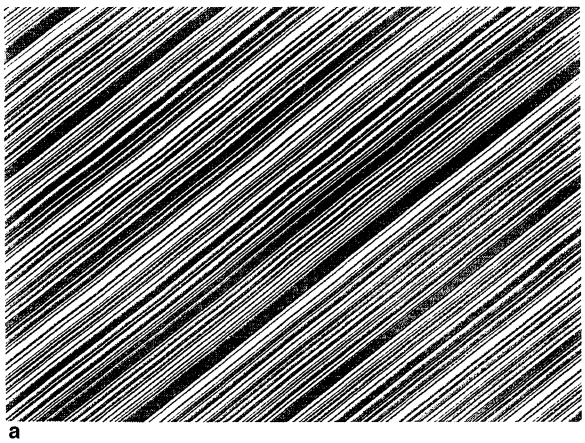
FIGURE 14 The density $P(1)$ for the Cellular Automaton of Figure 12.

evolving in time. They produce the same evolution pattern and the trajectories of the two systems coincide in state space. If, at a time step, a small damage (or defect) is added to one of them (i.e., a change in the state of one or more cells due to noise), then this damage may be quickly recovered or replicated without change (frozen damage), or spread. The Lyapunov exponent is the rate of the exponential divergence of the distance between the two initially coinciding trajectories in state space, in the limit of long times. Generally, if the Lyapunov exponent of a Cellular Automaton rule is negative then the damage is recovered, if the Lyapunov exponent is zero, or close to zero, the damage is frozen, and if the Lyapunov exponent is positive, the damage spreads. The larger the value of the Lyapunov exponent the faster the damage spreads. The Lyapunov exponents for some Cellular Automata rules have been calculated in [23]. If, in a hybrid Cellular Automaton, one of the rules has a positive Lyapunov exponent the existence of the another (or other) rule(s) may be considered as a damage which will spread with time, resulting in a large cycle length. In Figure 15a the evolution of a rule 42 Cellular Automaton for 450 time steps from a random initial state is shown. The Cellular Automaton consists of 501 cells, and periodic boundary conditions are imposed on it. The Lyapunov exponent of this Cellular Automaton is nearly zero. In Figure 15b a Cellular Automaton with the same configuration, and the same initial state is shown, but rule 18 is also activated on cell 350 to cell 450 and from time step 50 to time step 150. It is evident that the damage caused by the activation of rule 18 is frozen. In Figure 16a the evolution of a rule 106 Cellular Automaton is shown, with the same configuration and the same initial state as that of Figure 15. The Lyapunov exponent of this Cellular Automaton is 0.75. Figure 16b shows the same



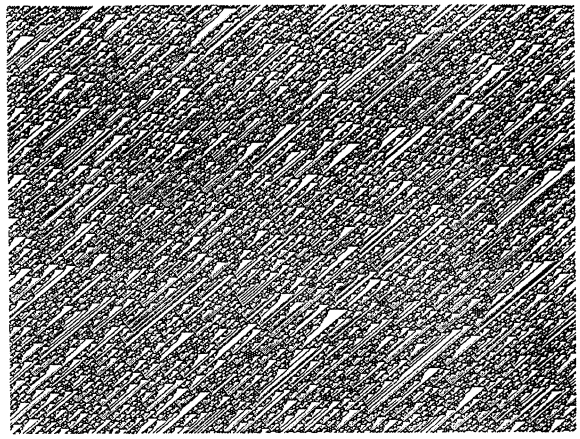
b

FIGURE 15 (Continued).



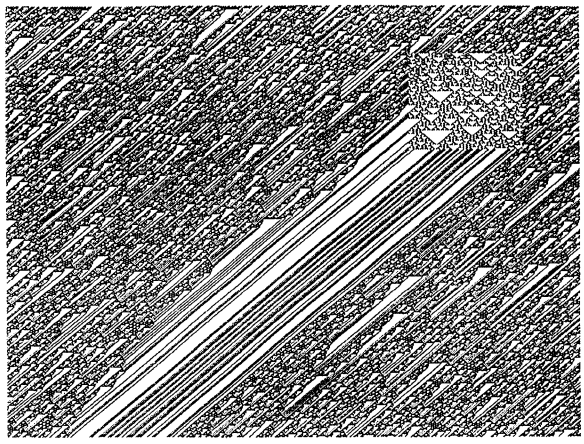
a

FIGURE 15 a) Evolution of a rule 42 Cellular Automaton. b) Evolution of the same Cellular Automaton, when the rule 18 is also activated.



a

FIGURE 16 a) Evolution of a rule 106 Cellular Automaton. b) Evolution of the same Cellular Automaton, when the rule 18 is also activated.



b

FIGURE 16 (Continued).

Cellular Automaton, but here the rule 18 is also activated on cell 350 to cell 450 and from time step 50 to time step 150. It is evident that the damage spreads with time resulting in a cycle length larger than that of the Cellular Automaton of Figure 16a.

Consider a 5-bit pseudorandom pattern generator. The five bits are generated by a rule 106 Cellular Automaton that consists of five cells. The state transition graph of this Cellular Automaton is shown in Figure 17. Each node represents a global state. Node 1 represents the global state 0 0 0 0 1, node 2 represents global state 0 0 0 1 0, node 3 represents global state 0 0 0 1 1 and so on. The arrows indicate the transitions of the global state. If the Cellular Automaton is found at state 4 (0 0 1 0 0), then, according to the graph, the next state is 8 (0 1 0 0 0) and the preceding state was 2 (0 0 0 1 0). If two states are not connected, transition from one to the other is impossible. The graph consists of three cycles, two of length 5 and one of length 15, which is the maximum cycle length of the Cellular Automaton.

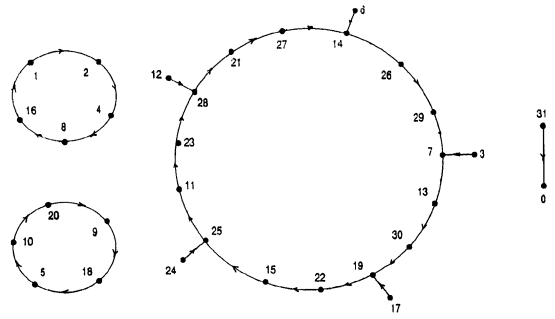


FIGURE 17 The state transition graph of a rule 106 Cellular Automaton with 5 cells.

Consider another 5 bit pseudorandom pattern generator. These five bits are generated by a hybrid in space-time Cellular Automaton that consists of five cells. The rule 106 is applied to all cells, but rule 18 can be activated on the second cell. The Cellular automaton evolves from initial global state 1 (0 0 0 0 1). The rules imposed on its five cells are: 106-106-106-106-106. The next global state is 2 and so on. After 4 time steps the global state is 16. At the fifth time step rule 18 is activated on the second cell for the next two time steps. Now the rules imposed on the five cells are: 106-18-106-106-106. Due to the activation of rule 18, the next global state of the Cellular Automaton is 9, the next 18 and the next 13. Now rule 18 is deactivated and the rules applied on the five cells are again: 106-106-106-106-106. The next state is 30, the next 19 and so on. The thick line in Figure 18 shows the evolution of the hybrid in space-time Cellular Automaton. Starting from an the initial global state 1 the sequence of global states is: 1-2-4-8-16-9-18-13-30-19-22-15-25-11-23-28-21-27-14-26-29-7-13.

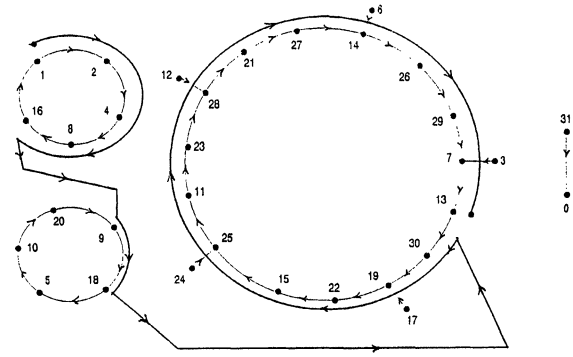


FIGURE 18 The thick line represents the state transition graph of a non-linear hybrid in space-time Cellular Automaton with 5 cells. The rule 106 is applied on all cells except for time steps 5, 6 and 7, in which rule 18 is applied on the second cell.

Global state 13 is reached for the second time, and the Cellular Automaton 1 enters in cycle. The cycle length of the non-linear hybrid in space-time Cellular Automaton is 23. Various cycle lengths can be achieved if rule 18 is activated at different time steps. Also different initial global states will lead to different cycle lengths.

5. CONCLUSIONS

It has been shown, in this paper, that Cellular Automata cannot produce random but pseudo-random patterns because of the correlation, both in space and time among the neighbouring cells. The spatial association between neighbouring cells has been expressed analytically and it holds for all Cellular Automata rules. The concept of hybrid in space-time Cellular Automata has been introduced, for the first time and it is in this paper suggested that non-linear hybrid in space-time Cellular Automata can be suitable for pseudorandom pattern generators for VLSI systems, because they can produce patterns with various densities $P(1)$ distributed at will in space and time. The cycle length of non-linear hybrid in space-time Cellular Automata depends on the initial global state and the time period in which the second rule is activated. Given a non-linear hybrid in space-time Cellular Automaton desirable cycle lengths can be achieved by proper choice of the initial global state and the time period in which the second rule is activated. Cycle lengths of non-linear hybrid Cellular Automata can only be estimated by extensive computer simulations. Evidence has been presented, that if one of the Cellular Automaton rules, forming the hybrid non-linear Cellular Automaton rule, has a large positive Lyapunov exponent, then it is highly probable that the hybrid non-linear Cellular Automaton will have a large cycle length.

Acknowledgements

The authors would like to thank the anonymous referees for their constructive criticism and help.

References

[1] Hortensius, P., McLeod, R., Pries, W., Miller, M. and Card, H. (1989). "Cellular Automata-Based Pseudorandom Number Generators for Built-In Self-Test", *IEEE Trans. on Comput. Aided Design*, **8**, p. 842.

[2] Hortensius, P., McLeod, R. and Card, H. (1989). "Parallel Random Number Generation for VLSI Systems Using Cellular Automata", *IEEE Trans. on Computers*, **38**, p. 1466.

[3] Tsalides, Ph., York, T. and Thanailakis, A. (1991). "Pseudorandom pattern generators for VLSI systems based on linear Cellular Automata", *Inst. Elect. Eng., Proc. pt E*, **138**, p. 241.

[4] Serra, M., Slater, T., Muzio, J. and Miller, M. (1990). "The Analysis of One-Dimentional Linear Cellular Automata and their Aliasing Properties", *IEEE Trans. on Computer-Aided Design*, **9**, p. 767.

[5] Hortensius, P., McLeod, R. and Card. H. (1990). "Cellular Automata Based Signature Analysis for Built-In Self-Test", *IEEE Trans. on Computers*, **39**, p. 1273.

[6] Das, A. and Chaudhuri, P. (1993). "Vector Space Theoretic Analysis of Additive Cellular Automata and its Application for Pseudoexaustive Test Pattern Generation", *IEEE Trans. on Computers*, **42**, p. 340.

[7] Damarla, T. and Sathaya, A. (1993). "Applications of one-Dimensional Cellular Automata and Linear Feedback Shift Registers for Pseudo-exaustive Testing", *IEEE Trans. on Computer-Aided Design of Integr. Cirq. and Systems*, **12**, p. 1590.

[8] Wolfram, S. (1986). "Random Sequence Generation by Cellular Automata", *Advances in Applied Mathematics*, **7**, p. 123.

[9] Gerhardt, M. and Schuster, H. (1989). "A Cellular Automaton describing the formation of spatially ordered systems", *Physica D*, **36**, p. 209.

[10] Gerhardt, M., Schuster, H. and Tyson, J. (1990). "A Cellular Automaton model of excitable media", *Physica D*, **46**, p. 392.

[11] Chopard, B. and Droz, M. (1991). "Cellular Automaton model for the Diffusion Equation", *Journal of Statistical Physics*, **64** , p. 859.

[12] Kong, X. and Cohen, E. (1991). "A kinetic theorist's look at lattice gas Cellular Automata", *Physica D*, **47**, p. 9.

[13] Toffoli, T. and Margolus, N. (1991). "Programmable matter: concepts and realization", *Physica D*, **47**, p. 263.

[14] Wolfram, S. (1983). "Statistical mechanics of Cellular Automata" *Reviews of Modern Physics*, **55**, p. 601.

[15] Wolfram, S. (1984). "Universality and complexity in Cellular Automata", *Physica*, **10D**, p. 1.

[16] Peitgen, H., Jurgens, H. and Saupe, D. (1992). "Chaos and Fractals" Springer-Verlag .

[17] Martin, O., Odlyzko, A. and Wolfram, S. (1984). "Algebraic Properties of Cellular Automata", *Commun. Math. Phys.*, **93**, p. 219.

[18] Pitsianis, N., Tsalides, Ph., Bleris, G., Thanailakis, A. and Card, H. (1989). "Deterministic One-Dimensional Cellular Automata", *Journal of Statistical Physics*, **56**, p. 99.

[19] Pitsianis, N., Bleris, G., Tsalides, Ph., Thanailakis, A. and Card, H. (1989). "Algebraic Theory of Bounded One-dimensional Cellular Automata", *Complex Systems*, **3**, p. 209.

[20] Knuth, D. (1981) "Seminumerical Algorithms", Addison Wesley.
[21] Yarmolic, V. and Demidenco, S. (1988) "Pseudorandom Sequences for Random Testing", Wiley.
[22] Wolfram, S. (1985). "Origins of Randomness in Physical Systems", *Phys. Rev. Lett.*, **55**, p. 449.
[23] Bagnoli, F., Rechtman, R. and Ruffo, S. (1992). "Damage spreading and Lyapunov exponents in Cellular Automata", *Physics Letters A*, **172**, p. 34.
[24] Vichniac, G. (1990). "Boolean Derivatives on Cellular Automata", *Physica D*, **45**, p. 63.
[25] Peitgen, H. O, Jurgens, H. and Saupe, D. (1992). "Chaos and Fractals", Springer-Verlag.

Authors' Biographies

Ioannis Karafyllidis received the Dip. Eng. degree in Electrical Engineering from Aristotle University of Thessaloniki, in 1983 and the Ph.D. degree from Aristotle University of Thessaloniki, in 1991. He is an Ass. Professor in the Department of Electrical and Computer Engineering, Democritus University of Thrace. His current research interests are VLSI physical design automation, VLSI systems and cellular automata theory and applications and simulation of VLSI technology processes. He is a member of the IEEE and the Technical Chamber of Greece (TEE).

Ioannis Andreadis received the Dip. Eng. Degree in Electrical Engineering from Democritus University of Thrace, in 1983, and the M.Sc and Ph.D. degree from UMIST in 1985 and 1989 respectively. He is an Ass. Professor in the Department of Electrical and Computer Engineering, Democritus University of Thrace. His current research interests are in machine vision and applications of cellular automata in image processing. He is a member of the IEEE and the Technical Chamber of Greece (TEE).

Philippos Tsalides received the Dipl. degree from the University of Padova, Italy in 1979 and the Ph.D. Degree from Democritus University of Thrace. He is a Professor in the Department of Electrical and Computer Engineering, Democritus University of Thrace. His current research interests include VLSI architectures, VLSI systems, BIST techniques, applications of cellular automata in image processing, as well as in computer systems. Dr. Tsalides is a member of the Technical Chamber of Greece (TEE).

Adonios Thanailakis received the B.Sc., degree in physics and electrial engineering from the University of Thessaloniki, Greece in 1964 and 1968, respectively, and M.Sc., and Ph.D. degrees in electrical engineering and electronics from UMIST, Manchester, England, in 1968 and 1971, respectively. He is a Professor in the Department of Electrical and Computer Engineering, Democritus University of Thrace. His current research activity includes amorphous materials and devices, photovoltaic conversion of solar energy, applications of group theory in physical and computational systems, and VLSI systems design. He has published a great number of scientific and technical papers, as well as four textbooks on materials, devices, electronic noise, and photovoltaic conversion of solar energy. Professor Thanailakis served as Rector (President) of the Democritus University of Thrace. He is a member of the Institute of Physics, the European Physical Society, the Technical Chamber of Greece, and of the Institute of Heliotechnics of Greece.