

# ANALYSIS OF ELEMENTARY CELLULAR AUTOMATA BOUNDARY CONDITIONS

K. Salman

Middle Tennessee State University  
Murfreesboro, TN 37132, USA  
[ksalman@mtsu.edu](mailto:ksalman@mtsu.edu)

## ABSTRACT

We present the findings of analysis of elementary cellular automata (ECA) boundary conditions. Fixed and variable boundaries are attempted. The outputs of linear feedback shift registers (LFSRs) act as continuous inputs to the two boundaries of a one-dimensional (1-D) Elementary Cellular Automata (ECA) are analyzed and compared. The results show superior randomness features and the output string has passed the Diehard statistical battery of tests. The design has strong correlation immunity and it is inherently amenable for VLSI implementation. Therefore it can be considered to be a good and viable candidate for parallel pseudo random number generation.

## KEYWORDS

*Linear Feedback Shift Registers, Cellular Automata, Boundary Conditions, Diehard*

## 1. INTRODUCTION

Both LFSRs and CAs have been used extensively in a wide area of applications, particularly random number generation for Mont Carlo simulation, communications, gaming, cryptography and network security, to name a few, [1-8]. LFSRs, albeit simple in structure and design were proven to have comparatively weak statistical features when utilized in the production of pseudo random numbers (PRNs) for cryptographic applications, [2]. The weakness can be attributed to the linearity of the *Exclusive-Or* function used in the feedback network. Additionally, non-linear feedback shift registers have their problems as well [1]. On the other hand, a uniform 1-D CA, where one rule is implemented throughout the spatiotemporal evolution of the CA, have shown unique and useful characteristics, and have been suggested by [3,4] and others for use in random number generation. A notable impediment however, is the input to the boundaries of the CA, where, for practical realization restrictions, it is confined to a limited span length. For example, a necessary condition for an unbounded (bi-infinite) 1-D ECA to produce a pseudo random contiguous string of length  $\xi \in N$  is to have a span of  $\xi + 2$  cells long. Hence, a relatively long string of output will render the CA overly unsuitable. However, a shorter span  $K \in N$  implies a constant span length and therefore a fixed and limited number of cells. Hence, inputs are needed to feed the two extremities of the ECA. One approach attempted to solve this problem is to make the ECA evolve in a continuous loop (referred to as autonomous or *periodic*), in which case the peripheral cells (i.e. the last and the first extreme cells) are made adjacent to each other, as

depicted in figure 1. An alternative technique used earlier in the literature is to feed the peripheral cells with fixed inputs. Figure 2 depicts the various boundaries used from  $GF(2)$ .

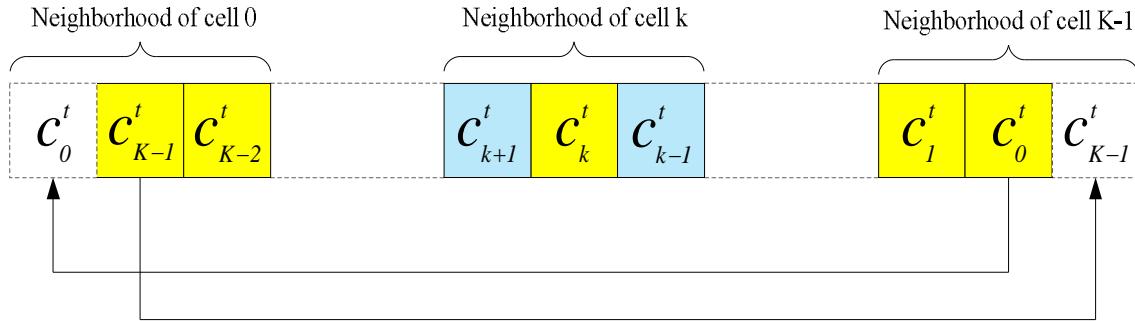


Figure 1, ECA classical *periodic* boundary configuration

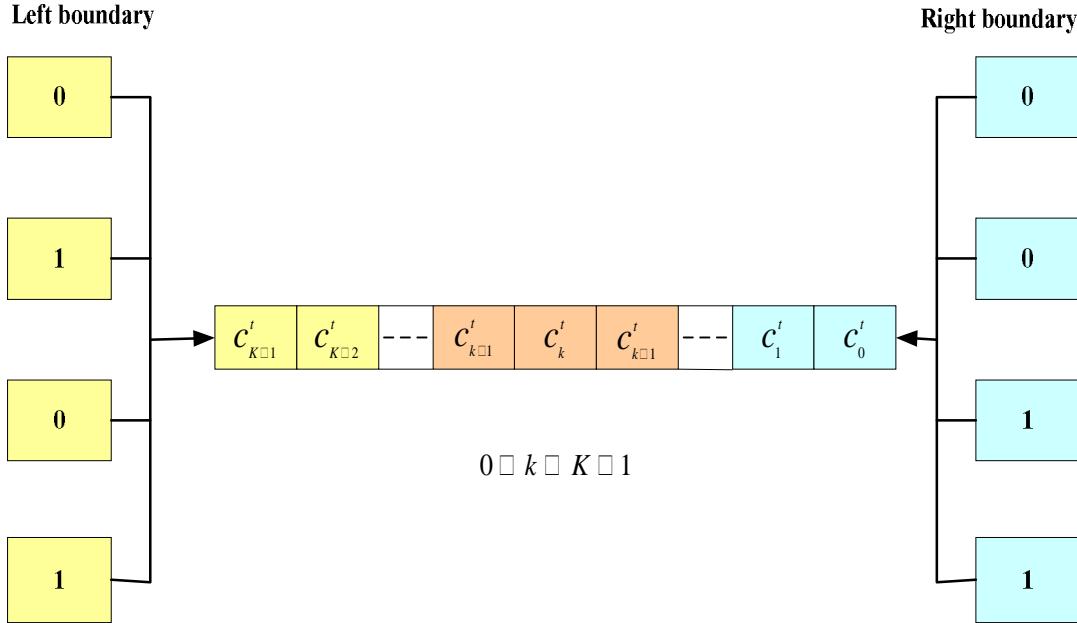


Figure 2, common elementary cellular automaton fixed boundary conditions.

All these methods running under, for example, chaotic rule 30 on uniform one-dimensional ECAs have produced much shorter periods than the LFSR and drastically failed the well-established Diehard battery of tests [7]. This paper reports the findings of such fixed boundary conditions as well as the findings of a new method whereby a pair of uncorrelated LFSRs are used to generate the two boundary conditions. With this design the output string of the ECA evolving for time steps  $T = 2^K$ , where  $K$  is the span length, has shown at least comparable results when subjected to the Diehard battery of tests and produced attractive parallelism and enhanced the asymptotic complexity. The whole ECA span length output has been used in contrast to the single cell output of the previously published PRN sequence of Wolfram, [4]. This paper is arranged such that the

theoretical analysis and the proposed approach are included in the section called Preliminaries, while the results section discusses the improvement in the performance of the ECA. The conclusion finalizes the outcome of the paper.

## 2. PRELIMINARIES

For the purpose of this paper we will restrict our attention towards one dimensional binary cellular automaton, henceforth referred to as Elementary Cellular Automaton (ECA). The cells are arranged on a linear finite lattice, with a symmetrical neighborhood of three cells and radius  $r = 1$ . Each cell takes its value from the set  $G = \{0, 1, \dots, p\}$  and since we defined the automaton as an ECA, it implies that  $p = 2$ . All cells are updated synchronously and the cells are restricted to local neighborhood interaction with no global communication. The ECA will evolve according to one uniform neighborhood transition function, which is a local function (rule)  $f: G^{2r+1} \rightarrow G$  where the ECA evolves after certain number of time steps  $T \in \mathbb{Z}$ . Out of a total of  $P^{p^{2r+1}}$  rules we use rule 30 as suggested by Wolfram and adopt his numbering scheme [3, 4]. It follows that a one dimensional (1-D) ECA is a linear register of  $K \in N$  memory cells. Each cell is represented by  $c_k^t$ , where  $k = [1: K]$  and  $t = [1, \infty)$ , that describes the content of memory location  $k$  at time evolution step  $t$ . Since  $p = 2$  then each cell takes one of two states from  $GF(2)$ . This implies the applicability of Boolean algebra to the design over  $GF(2)$ . A minimum Boolean representation of chaotic Rule 30 in terms of the relative neighborhood cells can be given by  $c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + c_{k-1}^t)$  or  $c_k^{t+1} = c_{k+1}^t + c_k^t + c_{k-1}^t + (c_k^t \cdot c_{k-1}^t) \text{ mod } 2$ , where  $2 \leq k \leq K - 2$ . Figure 3 depicts the first logic expression.

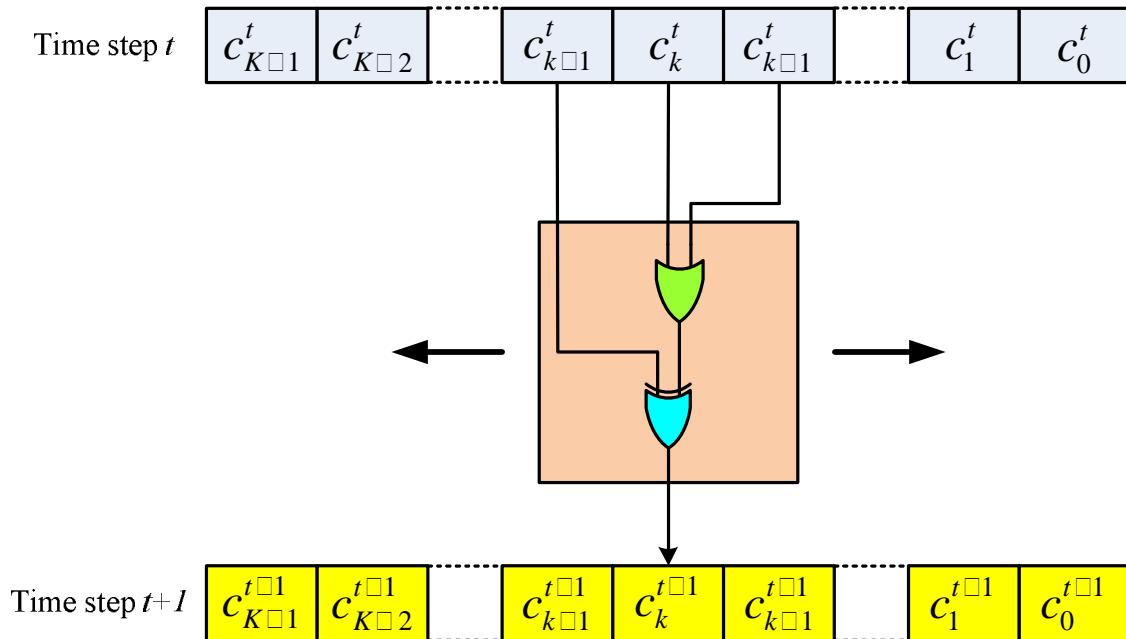


Figure 3, illustration of Rule 30 operating on the present state of neighborhood at time step  $t$  to produce the next state cell at time step  $t + 1$ .

Furthermore, since the ECA is actually a finite state machine then the present state of the neighborhood  $c_{k+1}^t, c_k^t, c_{k-1}^t$  of cell  $c_k^t$  at time step  $t$  and the next state  $c_k^{t+1}$  at time step  $t + 1$ , can be analyzed by the *state transition table* and the *state diagram* depicted in figure 4.

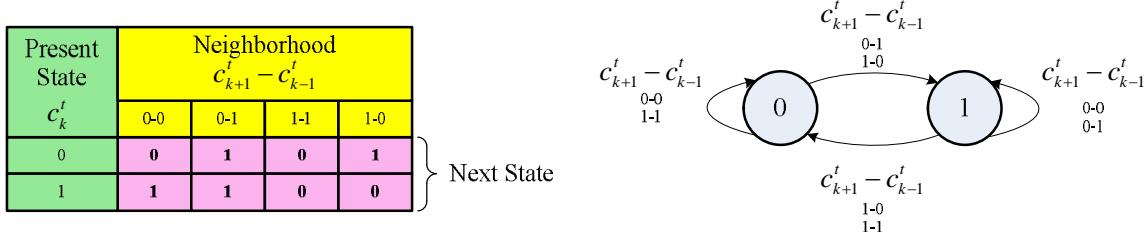


Figure 4, state machine analysis of Rule 30

It can be seen from above that in order to evolve from the present time step  $t$  to the next time step  $t + 1$ , each cell at lattice location  $k$  would require the present state of itself  $c_k^t$  as well as the present state of the other two cells in its neighborhood  $c_{k+1}^t$  and  $c_{k-1}^t$ . Therefore, if the ECA of span length  $K$  is allowed to expand freely leftwise and rightwise, as illustrated in figure 5, the number of cells required in one time step  $t + 1$  would be  $K + 2$ . Hence, the span length needed for time evolution steps  $T$  would be  $K + 2T$ . If the center cell of the seed is represented by  $c_k^0$  then the same cell will be represented by  $c_T^0$  after a total of  $T$  time steps.

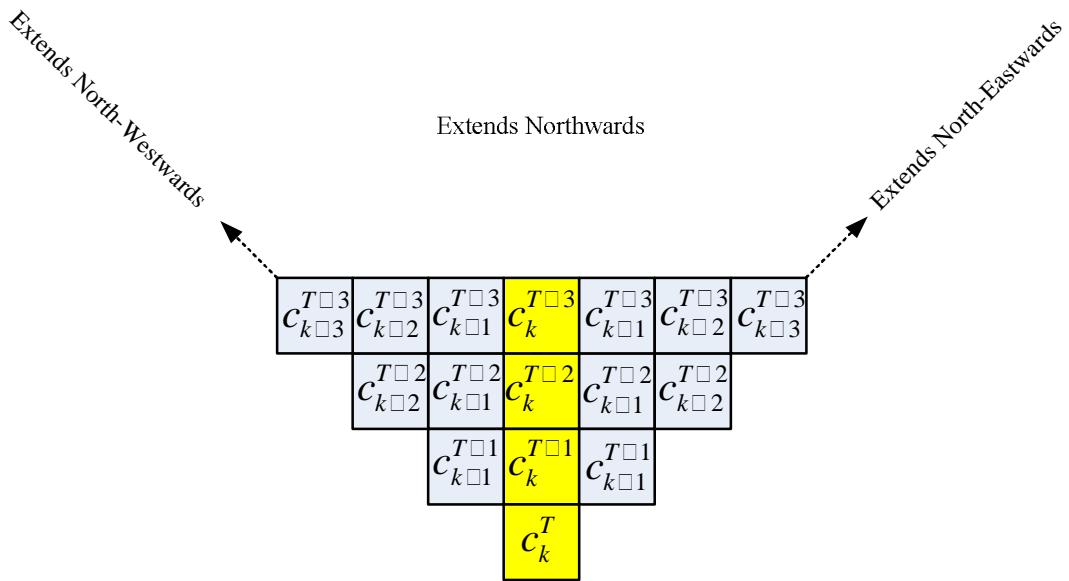


Figure 5, illustration of time evolution of a bi-infinite 1-D ECA

Admittedly, since the role of the PRN generator is to stretch a seed into a concatenated sequence of numbers with certain high entropy, it can be shown that a cellular automaton is a viable platform that can perform this task quite efficiently. As a rudimentary, albeit illustrative example, let a bi-infinite ECA (i.e. without boundary conditions) running under rule 30 be used to stretch a

seed  $S$  consisting of 13-bit,  $S = 01100100001 = 321_H$  by appending zeros to the least significant digits. As depicted in figure 6, one configuration of the output could be  $1011110010100011110110011$  or  $17947B3_H$  (when appending zeros at the least significant digits) by means of the concatenation scheme shown, or equivalently using bi-infinite ECA:  $321_H \xrightarrow{\text{rule 30}} 17947B_H$ . When optimum complexity of the output string is desired the center bit only can be selected according to the scheme suggested by Wolfram, [4], in which case the size of the output string would be reduced drastically. In this case the output string would be  $10001$  or  $11_H$ , when appending zeros at the least significant digits, i.e.  $321_H \xrightarrow{\text{rule 30}} 11_H$ .

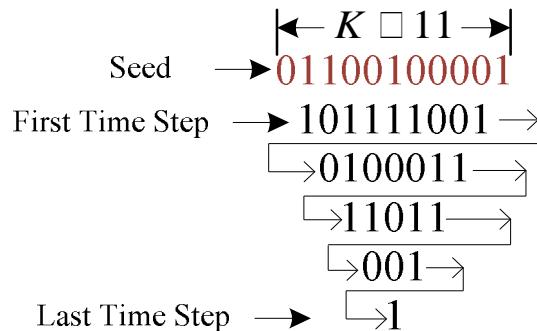


Figure 6, Simple time evolution of an unbounded 1-D ECA under GF (2).

It follows that if the ECA is unbounded then for a string of  $T$ -bits formed by the concatenation of the center cell would require a seed of span length  $2T + 1$  as can be deduced from the two figures. This condition will eventually lead to an unpractical span of the ECA. Hence, it is imperative that the ECA has to be bounded. The open literature is rich with research on fixing the size of the ECA and provides data for the extreme cells of the bounded ECA. Figure 2 gives a brief account of some common fixed boundary conditions. Figure 7 categorizes the boundary conditions to include the new boundary condition proposed in this paper using LFSR as a new source for boundary conditions. The miscellaneous category includes either some ad hoc permutations of the fixed boundaries or some fixed sequence of inputs. The autonomous category, commonly referred to as *periodic*, makes the extreme cells of the ECA adjacent, as illustrated in figure 2. The resultant ECA becomes circular as depicted in figure 8, and with time evolution it can be visualized as a cylinder. The expression for the extreme left and right cells at time step  $t + 1$  are respectively  $c_{K-1}^{t+1} = c_0^t \oplus (c_{K-1}^t + c_{K-2}^t)$  and  $c_0^{t+1} = c_{K-1}^t \oplus (c_1^t + c_0^t)$ .

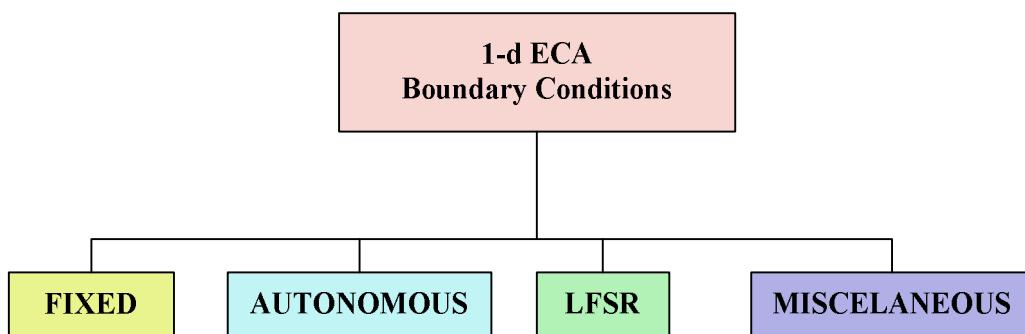


Figure 7, Categorization of a fixed span 1-D ECA boundary condition sources.

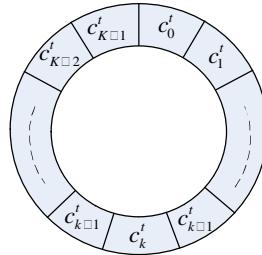


Figure 8, another illustration of the autonomous (periodic) boundary conditions.

The proposed approach of using LFSRs as new source of boundary conditions is illustrated in figure 9. The next state of all cells of the ECA except the end cells is represented, for rule 30, by  $c_k^{t+1} = c_{k+1}^t \oplus (c_k^t + c_{k-1}^t)$  for  $2 \leq k \leq K - 2$  while the next state of the two extreme cells are represented by  $c_{K-1}^{t+1} = L_0^{t+1} \oplus (c_{K-1}^t + c_{K-2}^t)$  for the left hand cell while  $c_0^{t+1} = c_1^t \oplus (c_0^t + R_0^{t+1})$  represents the next state of the right hand cell. Each of the two LFSRs have taps derived from a set of  $\phi(L - 1)$  primitive polynomials, where  $\phi$  is *Euler's totient* function. It can be seen that the span lengths of the two registers can be different as well as the choice of the primitive polynomials. When the spans are different then different totient functions will be derived. The size of the registers spans and consequently the Euler's totient functions can play a significant role in enhancing the complexity of the output pseudo random sequences derived from the elementary cellular automaton. It can be seen that for a total of  $T \in \mathbb{Z}$  time evolution of the ECA, the complexity of the output string is enhanced by a factor of  $O(T^2 * \lceil \log_2 T \rceil * (\phi(\lceil \log_2 T \rceil))^2)$  where  $\phi(\lceil \log_2 T \rceil)$  is the totient function of each. The above expression assumes each LFSR is constructed from a number of memory elements  $N = \lceil \log_2 T \rceil$  for maximum complexity enhancement.

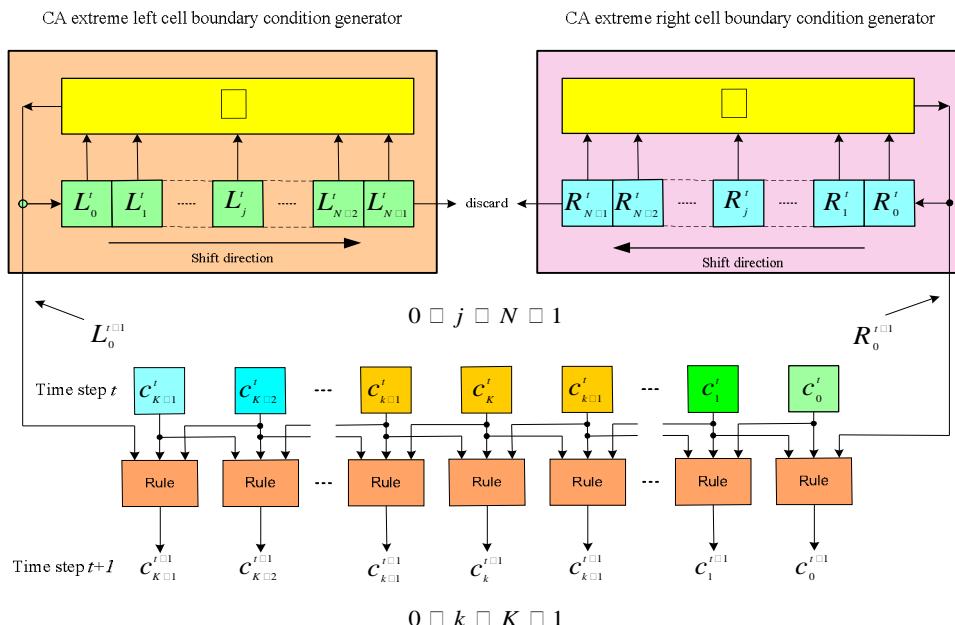


Figure 9, Block Diagram Representation of the Proposed ECA System, reversing the order of indexing, such that the most significant cell is the extreme left hand cell and vice versa for the extreme right hand cell which becomes the least significant cell.

### 3. ECA RULE SPACE

In order to test the statistical properties of the new proposed design, we developed a suite of programs emulating the types of boundary conditions as classified in figure 7 for a range of spans for both the ECA and the LFSRs used as boundaries. We will include snapshots of results obtained for representative runs on the Diehard battery of tests [8], which has been adopted in this paper due to its well established stringent requirements on the statistical randomness of the output string. The rule space of the 1-D ECA where the alphabet  $p \in GF(2) = 2$ , consists of  $p^{p^{2r+1}} = 2^{2^3} = 256$  distinct rules. The numbering scheme adopted here is attributed to Wolfram, [3], so is the classifications of the rule space. Although Wolfram's ECA rule classification is essentially based on the phenomenological observations of the space-time images it seems to be quite effective. Many other scholars researched the classification but their findings more or less agree with Wolfram's classification. The rule space is divided roughly into four classes:

- I. Evolution leads to homogeneous fixed points.
- II. Evolution leads to periodic configurations.
- III. Evolution leads to chaotic, aperiodic patterns.
- IV. Evolution produces persistent, complex patterns of localized structures.

Class III, the chaotic rules, is the only class that can be useful for pseudo random number, when used in a uniform or homogeneous configuration. The other three classes produce too short periodic patterns for such applications. Since the output string of any pseudo random number generator has to satisfy certain well established conditions, such as uniformity and very long periods, the rule space can be reduced to a small sub-space. For example, in order to satisfy the uniformity condition, i.e. the density of the zeros and ones in the output sequence have to be asymptotically the same, it follows that the minterms of the chaotic rules must be balanced. This means that the number of minterms that are asserted must equal the number of minterms that are deasserted, see figure 4. Consequently the rule space will be reduced to a sub-space of size  $\frac{2^3!}{4!(2^3-4)!} = 70$ . Furthermore, it has been found that sixteen rules out of this number are actually chaotic and can tentatively be considered suitable for pseudo random number generation. These rules are: 30, 45, 60, 75, 86, 89, 90, 101, 102, 105, 135, 149, 150, 153, 165, 195. Some of these rules are *linear* where the logical expression of the rule depends only on the linear primitive, i.e. the Exclusive-Or primitive (mode 2). These are rules 60, 90, 102, 105, 150, 153, 165, and 195. The remaining eight rules are: 30, 45, 75, 86, 89, 101, 135, and 149. These two groups can be further grouped into *equivalence* rules, [9], producing the following sub-groups with their group leaders:

1. Non-linear group 30 consisting of the four rules: 30, 86, 135 and 149
2. Non-linear group 45 consisting of the four rules: 45, 75, 89 and 101
3. Linear group 60 consisting of the four rules: 60, 102, 153, 195
4. Linear group 90 consisting of the two rules: 90, 165
5. Linear group 105 consisting of the two rules: 105, 150

Rule 86 is the reflection of rule 30 while rule 149 is the reflection of rule 135. Also rule 135 is the negation of rule 30. Likewise with group 45, rule 101 is the reflection of rule 45 while rule 89 is the reflection of rule 75. The dynamical behavior of the rules within any group is statistically similar although the time-space diagrams can be different in reflection or negation. Examples will be given to highlight this feature. It is apparent that when the rule is used in a uniform

configuration, i.e. using just one rule throughout the ECA time evolution, the linear rules will have lower complexities than the non-linear rules, and therefore will be less favorable for strong pseudo random number generation. We will, therefore pay more attention to the two non-linear groups and concentrate the efforts on the two rule leaders 30 and 45.

## 4. SAMPLES

The following will show the results of running the ECA on the two rules 30 and 40 for a variety of span length  $K$ , time evolution  $T$ , representative sample of seeds and the different boundary conditions quoted in figures 2 and 7.

### 4.1. Fixed Boundary Conditions

The space-time images shown in tables 1 and 2 illustrate the dynamical behavior of sub-group rule 30 for span length  $K = 15$  and time evolution  $T = 60$  with fixed boundary conditions 00, 01, 10 and 11. The initial seed consists of the center cell only being asserted while the rest are deasserted. The reflection property of rule 30 and 86 as well as rule 135 and rule 149 is quite obvious in the images. The negation feature is also apparent when examining the self similar shapes (or fractals), as these are being filled with the complement color. Another observation is that while the whole rule 30 sub-group belongs to class III, which is *chaotic*, the space-time patterns show that the cellular automaton evolves to class I, which is *fixed point*, when the two fixed boundaries are 00. It also evolves to class I when the two boundaries are similar and 11 for rules 135 and 149. However, when the fixed boundaries are dissimilar, the ECA evolves to class II, which is *periodic configurations*, with a short period of 2. The same is true for rules 30 and 86 when the boundaries are 11. When the seed is changed to random input, most of the features captured in the previous tables 1 and 2 are repeated in tables 3 and 4. For example the ECA also exhibits evolution from rules that belong to class III to class I dynamics when the two boundaries are the same, i.e. 00 or 11. The dissimilar boundaries, i.e. 01 or 10 transforms the ECA into class II dynamics with very short periods. It is also observable that the reflection property is not as clear and identical as was the case with the center seed. It should be noted here that the behavior of the ECA can be very sensitive to the seed. For example the ECA with repetitive patterns of 01 in the seed will keep the ECA in the same state if the boundaries are 11. There are other seed patterns that cause either similar action or produce very short periods and transform the ECA into class II. The same approach has been attempted with rule 45 sub-group. The results are displayed in tables 5-8. Note that the same random seed was used for all the four fixed boundaries. Similar conclusions can be drawn with rule 45 sub-group. These rules also exhibit self similar (fractal) patterns but they are different from those encountered with rule 30 sub-group. The reflection property is again quite observable in the patterns of rule 45 and its reflection rule 101 as well as rule 75 and its reflection rule 89 when the fixed boundary conditions are the same, i.e. 00 or 11. It is also easy to observe that when the ECA under the rules of sub-group 45 is transformed into class II, the period is generally longer than those encountered with sub-group 30. This does not mean that sub-group 45 can produce better random results than sub-group 45. The inverse is actually true and will be observed when the ECA is run in the periodic configurations or with LFSR boundary conditions. Rule 30 has always shown somehow better random sequences than rule 45 and its sub-group. In summary, both groups can evolve to class I or II with short periods when the ECA is subjected to fixed boundary conditions. It is obvious that the dynamics of the ECA with the fixed boundary conditions, under all the different fixed boundary conditions cannot make such configurations viable for pseudo random number generation. Therefore running the

ECA with these boundary conditions for a longer time evolution will not produce any useful results. Hence, the ECA under these conditions cannot be of use for the applications in consideration in this paper.

Table 1

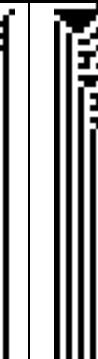
Rule 30 sub-group Center Seed $K = 15, T = 60$							
Fixed Boundary Conditions							
00				01			
Rule 30	Rule 86	Rule 135	Rule 149	Rule 30	Rule 86	Rule 135	Rule 149
							

Table 2

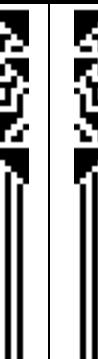
Rule 30 sub-group Center Seed $K = 15, T = 60$							
Boundary Conditions							
10				11			
Rule 30	Rule 86	Rule 135	Rule 149	Rule 30	Rule 86	Rule 135	Rule 149
							

Table 3

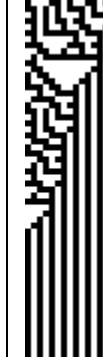
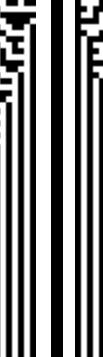
Rule 30 sub-group Random Seed $K = 15, T = 60$							
Boundary Conditions							
00				01			
Rule 30	Rule 86	Rule 135	Rule 149	Rule 30	Rule 86	Rule 135	Rule 149
							

Table 4

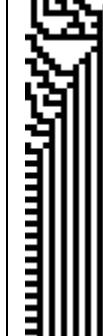
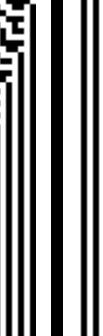
Rule 30 sub-group Random Seed $K = 15, T = 60$							
Boundary Conditions							
10				11			
Rule 30	Rule 86	Rule 135	Rule 149	Rule 30	Rule 86	Rule 135	Rule 149
							

Table 5

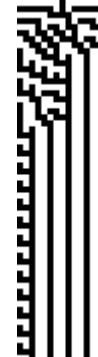
Rule 45 sub-group Center Seed $K = 15, T = 60$							
Fixed Boundary Conditions							
00				01			
Rule 45	Rule 101	Rule 75	Rule 89	Rule 45	Rule 101	Rule 75	Rule 89
							

Table 6

Rule 45 sub-group Center Seed $K = 15, T = 60$							
Fixed Boundary Conditions							
10				11			
Rule 45	Rule 101	Rule 75	Rule 89	Rule 45	Rule 101	Rule 75	Rule 89
							

Table 7

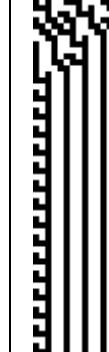
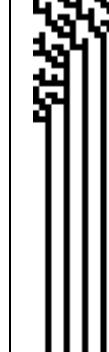
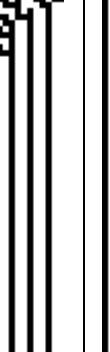
Rule 45 sub-group Random Seed $K = 15, T = 60$							
Fixed Boundary Conditions							
00				01			
Rule 45	Rule 101	Rule 75	Rule 89	Rule 45	Rule 101	Rule 75	Rule 89
							

Table 8

Rule 45 sub-group Random Seed $K = 15, T = 60$							
Fixed Boundary Conditions							
10				11			
Rule 45	Rule 101	Rule 75	Rule 89	Rule 45	Rule 101	Rule 75	Rule 89
							

#### 4.2. Periodic and LFSR Boundary Conditions

Using the same initial random seed and for the same span length and time evolution, i.e.  $K = 15, T = 60$ , the ECA was run under the two sub-groups of rules 30 and 45 and the space-time images of these runs are displayed in tables 9 and 10, respectively. As previously observed, the negation property is quite noticeable in contrast to the reflection property. For example, rule 30 and its reflection rule 86 do not show clearly the action of reflection. However, the two rules

show different outputs as far as the patterns are concerned. The same thing applies to the ECA bounded by the LFSR inputs. The span length of the LFSR uses was just 7-bit producing a maximum cycle length of  $2^7 - 1 = 127$  bit. Although the seed and the rule are the same but the space-time images are quite different. It should be clear that the difference is due to the effect of the boundary inputs that will propagate at the full speed, or *speed of light*, as some scholars would like to call. As the ECA evolves more inputs are injected into the ECA making the effort of reckoning the inputs from the LFSR an exhaustive process. The data was subjected to a cycle catcher program and it was found that all these runs did not show any repeated cycles even for time evolution of = 100 . The ECA was then run for longer time evolution and different span lengths. The amount of data collected for each run has to comply with the requirement of the test suite. The Diehard battery of test suite was selected in this research due to its popularity and stringent tests in academia and beyond. The amount of data required should exceed 80Mbit. The Diehard suite package transforms the data into binary data and then run the test suite on the binary data for the 15 tests. The last three tables, 11, 12 and 13 show the findings of the test suite. Table 11 shows the results for the periodic boundary conditions and for span lengths, 32, 33, 64, 128, 256 and 512 bit. The best results are, as expected with the longest span length of 512 although it passed 8 out of a total of 15 tests. Some of the tests could not be passed, e.g. tests 2, 4 and 8. Table 12 shows the results of running the ECA with LFSR boundaries. The two extreme boundary cells were generated from two different and uncorrelated LFSRs of span length 15-bit each. The improvement is clear. The ECA have passed 10 tests at a span length of 256-bit and failed 5 tests as compared to the periodic boundary conditions that passed 9 and failed 6. It is even better than the results of the periodic boundary conditions with double the size of the span length, 512-bit. The LFSR span lengths were increased such that they match the span length of the ECA. This configuration had shown superior results and new configuration have passed all the tests even down at span length of just 27-bit long. The results are displayed in table 13. The space-time images of the LFSR bounded ECA still show the standard rule 30 fractals but the effect of the perturbations of the boundaries from the large cycles of the two LFSRs is expected to have reduced the correlations of the cites. The output data is composed of the concatenation of all the states of the cellular automaton during its entire time evolution. It seems that the injection of the uncorrelated inputs from the two uncorrelated LFSRs cause to break the correlation that normally exists due to the local action of the local rule. Each bit from a boundary entry will propagate at full speed and meet the effect of the propagation of the other boundary at half the ECA span length. As there is a continuous stream of uncorrelated data being injected the global dependence of the cells will be highly reduced. However, the local dependence will continue to show in the shape of the usual fractals. It is expected that the complexity enhancement achieved in this configuration will make the design a viable scheme for the generation of cryptographically strong pseudo random numbers.

Table 9

Rule 30 sub-group Random Seed $K = 15, T = 60$							
Periodic Boundary Conditions				LFSR boundaries			
Rule 30	Rule 86	Rule 135	Rule 149	Rule 30	Rule 86	Rule 135	Rule 149
							

Table 10

Rule 4 sub-group Random Seed $K = 15, T = 60$							
Periodic Boundary Conditions				LFSR boundaries			
Rule 45	Rule 101	Rule 75	Rule 89	Rule 45	Rule 101	Rule 75	Rule 89
							

Table 11, Diehard tests results for 1-D ECA of variable spans and with autonomous boundaries.

P_VALUES	<b>S32</b>	<b>S33</b>	<b>S64</b>	<b>S128</b>	<b>S256</b>	<b>S512</b>
<b>T_1</b>	0.4913	0.6089	0.4871	0.5683	0.5976	0.7166
<b>T_2</b>	1	1	1	1	1	1
<b>T_3</b>	0.759	0.7895	0.5035	0.4525	0.643	1
<b>T_4</b>	1	1	1	1	1	1
<b>T_5</b>	1	1	0.4973	0.5195	0.5777	0.7068
<b>T_6</b>	1	1	0.804	0.7769	0.4856	0.7756
<b>T_7</b>	1	1	0.999	1	1	1
<b>T_8</b>	1	1	1	1	1	1
<b>T_9</b>	1	1	0.6235	1	0.431	0.3777
<b>T_10</b>	1	1	0.4376	0.6587	0.489	0.5068
<b>T_11</b>	1	1	0.5549	0.5016	0.457	0.4066
<b>T_12</b>	1	1	1	0.019	1	1
<b>T_13</b>	0.3985	0.4106	0.337	1	0.2194	0.375
<b>T_14</b>	1	1	1	0.3576	1	1
<b>T_15</b>	1	0.8809	1	1	0.8697	0.8524
<b>Summary</b>	<b>3 pass 12 fail</b>	<b>4 pass 11 fail</b>	<b>8 pass 7 fail</b>	<b>8 pass 7 fail</b>	<b>9 pass 6 fail</b>	<b>8 pass 7 fail</b>

Table 12, Diehard tests results for 1-D ECA of variable spans and with two LFSRs as boundaries of span 15-bit each.

P_VALUES	<b>S27</b>	<b>S28</b>	<b>S29</b>	<b>S30</b>	<b>S64</b>	<b>S128</b>	<b>S256</b>
<b>T_1</b>	0.8893	0.5869	0.6834	0.005	0.2502	0.5655	0.5638
<b>T_2</b>	1	1	1	1	1	1	1
<b>T_3</b>	0.402	0.2845	0.52	0.485	0.681	0.0795	0.144
<b>T_4</b>	0.997	1	1	1	1	1	1
<b>T_5</b>	1	1	1	1	0.4418	0.5753	0.4226
<b>T_6</b>	1	1	1	1	0.8211	0.6235	0.7855
<b>T_7</b>	1	1	1	1	0.8848	0.992	1
<b>T_8</b>	1	1	1	1	1	1	1
<b>T_9</b>	1	1	1	1	0.4273	0.0373	0.5168
<b>T_10</b>	1	1	1	1	0.2837	0.00035	0.0049
<b>T_11</b>	1	1	1	1	0.2291	0.1859	0.0362
<b>T_12</b>	1	1	1	1	1	1	1
<b>T_13</b>	1	0.0537	0.6473	0.4943	1	0.1131	0.0442
<b>T_14</b>	1	1	1	1	1	1	1
<b>T_15</b>	1	1	1	1	1	1	0.9056
<b>Summary</b>	<b>3 pass 12 fail</b>	<b>3 pass 12 fail</b>	<b>3 pass 12 fail</b>	<b>3 pass 12 fail</b>	<b>8 pass 7 fail</b>	<b>9 pass 6 fail</b>	<b>10 pass 5 fail</b>

Table 13, Diehard tests results for ECA of variable spans and with 2LFSRs for boundaries of same as the ECA spans.

P_VALUES	S27	S28	S32	S64	S128
<b>T_1</b>	0.242	0.43	0.3046	0.2398	0.2695
<b>T_2</b>	0.0744	0.4376	0.1128	0.5284	0.2123
<b>T_3</b>	0.8442	0.6365	0.3417	0.3317	0.5543
<b>T_4</b>	0.4688	0.47	0.4323	0.2713	0.0628
<b>T_5</b>	0.52235	0.4697	0.5166	0.4421	0.5454
<b>T_6</b>	0.4755	0.32	0.5486	0.5584	0.4654
<b>T_7</b>	0.6092	0.485	0.3151	0.4642	0.6849
<b>T_8</b>	0.5581	0.5083	0.4601	0.5009	0.6135
<b>T_9</b>	0.2253	0.6181	0.6947	0.5722	0.5413
<b>T_10</b>	0.8818	0.2469	0.9452	0.728	0.0897
<b>T_11</b>	0.7111	0.3404	0.1944	0.7524	0.5147
<b>T_12</b>	0.456	0.423	0.9646	0.9847	0.1522
<b>T_13</b>	0.3026	0.1387	0.2413	0.1063	0.3202
<b>T_14</b>	0.2085	0.6276	0.1753	0.3521	0.4801
<b>T_15</b>	0.343	0.5539	0.7578	0.4428	0.4845
<b>Summary</b>	<b>15 pass 0 fail</b>				

## 5. CONCLUSIONS

The string of contiguous stream data collected from evolution of the 1-D ECA for the center cell of various boundary conditions were tested by the 15 Diehard battery of tests. The various fixed boundary conditions failed the diehard tests almost completely and were considered unworthy reporting. The autonomous boundary conditions (i.e. periodic) have shown far better statistical properties than the fixed boundary conditions. However, it still falls far below the minimum requirements of the diehard tests for reliable considerations in producing dependable pseudo random numbers even for long spans of the ECA (512-bit). When the boundaries were fed from LFSRs results did not improve significantly until the span of the LFSRs were comparable to that of the ECA. The results steadily improved up to the upper bound when the two spans were comparable. It can be concluded that the new approach can produce good pseudo random numbers even at modest size of the ECA (i.e. 27-bit). More in depth study of the results show that the new approach produced superior *p*-values than the best of the autonomous results. It is easy to expect that the fixed boundary conditions cause an ECA running under Rule 30, which is in group III (i.e. the chaotic class) to evolve into Group I or II (i.e. point attractors or limit cycles with extremely small periods), according to Wolfram's ECA classification, [4]. Therefore, such boundary conditions preclude these ECAs from achieving strong pseudo random number generators. The autonomous (*periodic*) boundary conditions, on the other hand gave better results which is indicative of better distribution during ECA evolution. However, the periods of this type were far lower than the maximum length obtainable from LFSRs. The proposed design has an added favorable feature when considering the initial seeds. It is clear that all the possible  $2^K$  *K*-tuples can be used as seeds including the all 0's and all 1's that usually yield quiescent states.

This is not possible with any other known boundary conditions including the autonomous type. All the tests were performed using a single one as the initial seed. This is admittedly not the case in a practical situation. Some patterns were observed during the initial evolution of the ECA but did not persist. Although these initial patterns did not negatively impact the diehard tests it was found that avoiding the use of trinomials for the LFSRs and replace them with primitive polynomials of better distribution of the coefficients managed to remove these patterns. One salient feature of the design is the almost total destruction of the cross-correlation between different cells. This strong correlation is an inherent feature of LFSRs that can be observed as maximum and constant between any two cells of the LFSR and as linear patterns on the diagonal ridge between the outputs of the LFSR cells. An immediate consequence is the ability to use the ECA as a parallel source of pseudo random numbers that can be considered a strong candidate for parallel data compaction (signature analysis) in VLSI testing [8]. This is justified since the structure as depicted in figure 9 presents a simple memory-based and inherently parallel design that is amenable to large scale integration. Inspection of rule 30 reveals that the function is surjective. Since reversibility implies bijection, it follows that the proposed system is not clear cut reversible. Hence analytical techniques may not be available to adequately and inversely describe the spatiotemporal data evolution in at most polynomial time. For a LFSR of span  $S$ , there are  $2^N - 1$   $N$ -tuple words as seeds. The two LFSRs are uncorrelated and running independently and synchronously, hence the effective input computational complexity from these registers to the ECA would be  $2^N - 1^2$ . The 1-D ECA of span  $K$  can be initialized with a total of  $2^K$   $K$ -tuple words as initial seeds. There are a total of  $2^{2^3}$  rules, which is the rule space of a 1-D ECA. Thus the computational asymptotic complexity of the system is  $O((2^N - 1)^2 * 2^{2^3} * 2^K) \cong O(2^{3K})$  for  $K \cong N$ , as compared to  $2N$  for the LFSR and  $O(2^K)$  for a 1-D ECA with autonomous boundary conditions.

## REFERENCES

- [1] SIEGENTHALER, T. : ‘Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications’, IEEE Transactions on Information Theory, Vol. IT-30, No. 5, September 1984, pp. 776-780.
- [2] GUSTAVSON, F. G.: ‘Analysis of the Berlekamp-Massey Linear Feedback Shift-Register Synthesis Algorithm.’ IBM J. Res. Dev. **20**, Number 3, pp. 204-212, 1976.
- [3] WOLFRAM, S.: ‘A New Kind of Science’. Champaign, IL: Wolfram Media, 2002.
- [4] WOLFRAM, S.: ‘Random Sequence Generation by Cellular Automata’, Advances in Applied Mathematics. Volume 7, Issue 2, June 1986, Pages 123-169.
- [5] SEREDYNISKI, FRANCISZEK, BOUVRY PASCAL, and ZOMAYA, ALBERT Y.: ‘Cellular automata computations and secret key cryptography’, Parallel Computing, Vol. 30, 2004, pp. 753-766.
- [6] LLACHIINSKI, Andrew: ‘Cellular Automata: A Discrete Universe’, World Scientific, 2001, pp. 94.
- [7] HORTENSIUS, P.D., McLEOD, and CARD, H.C.:‘Parallel Random Number Generation for VLSI Systems Using Cellular Automata’, IEEE Transactions on Computers, Vol. 38, Issue 10, October 1989, pp. 1466-1473.
- [8] ‘The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness’, Florida State University, <http://i.cs.hku.hk/~diehard/>
- [9] Andrew Wuensche, and Mike Lesser, “The Global Dynamics of Cellular Automata”, Reference Volume I, Addison Wesley Publishing Company, 1992, ISBN: 0-201-55740-1.