Design of a Hybrid Programmable 2-D Cellular Automata Based Pseudo Random Number Generator

Dinakaran P, Gethzi Ahila Poornima I, Paramasivan B

***Abstract*: This paper proposes a hybrid programmable two-dimensional Cellular Automata (CA) based pseudo-random number generator which includes a newly designed rule set. The properties and evolution of one and two dimensional CA are revisited. The various metrics for evaluating CA as a Pseudo-Random Number Generator (PRNG) are discussed. It is proved that the randomness is high irrespective of the initial seed by applying this newly designed rule set. The PRNG is tested against a popular statistical test called Diehard test suite and the results show that the PRNG is highly random. The chaotic measures like entropy, hamming distance and cycle length have been measured**

***Keywords* : Cellular Automata, Pseudo-Random Number Generator, Entropy, Hamming Distance, Programmable CA.**

I. INTRODUCTION

A Cellular Automata (CA) is a model that has been studied in several branches of theory and occupied its unique identity in those fields. The field areas are physical theory, complexity science, mathematics, computational science, microstructure and biology modeling. A cellular automaton consists of cells as like finite array. Each cell is a finite state machine C = (Q, f) where Q is a finite set of states and f is a mapping f: QN → Q. The mapping f is called the local transition function. N is the number of cells the local transition function depends on. Each cell updates itself concerning the function 'f' at each iteration. Initially the CA was invented by Von Neumann. In the 1960s CA was studied as a particular type of dynamic system and the connection with the mathematical field of symbolic dynamics was established for the first time. In the 1980's Stephen Wolfram, worked on CA and found that CA is the best for the application of Pseudo-Random Number Generation (PRNG). The paper is organized as follows: In section I, the basic terminologies related to CA and the classifications of CA are presented.In Section II, the background of this work is explored. In section III, some works related to cellular automata and CA based PRNG are explored. In section IV, the proposed architecture is presented and related explanations are explained. Section V gives the analysis of results and comparison with the previous works. Section VI concludes the proposed work and suggests a future enhancement to be undertaken.

Manuscript received on March 15, 2020.
Revised Manuscript received on March 24, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author

Dinakaran P*, PG Scholar, National Engineering College, Kovilpatti, Tamilnadu, India. Email: dinase32@gmail.com

Gethzi Ahila Poornima I, Ph.D. Scholar, National Engineering College, Kovilpatti, Tamilnadu, India. Email: gethzi.akila@gmail.com

Paramasivan B, Professor, National Engineering, Kovilpatti, Tamilnadu, India. Email: bparamasivan@yahoo.co.in

II. BACKGROUND

Cellular Automata is composed of cells. Each of those cells contains a finite state machine called 'automaton'. In the cellular space, all the cells go from their current state to the next state according to the 'local rule'. These rules govern the transition between states. It is called 'local' because it only uses the states of the neighborhood and the current cell as its input. The local transition function is defined as f: Sn→S where n is the size of the neighborhood. For 2-state 3-neighborhood CA, the evolution of the ith cell can be represented as a function of the present states of (i-1)th, ith, and (i+1)th cells as:

$$xi(t+ 1) = f\{x1(t),\dots xn(t)\} \quad \text{--------} \quad (1)$$

where the function f represents the combinational logic. The cells surrounding a cell influence its next state. The neighborhood cells, one is leftmost of the current cell and other is rightmost to the current cell will states the behavior of the cellular space. The configuration is an instantaneous description or a snapshot of all cell states, representing a single point in time. A configuration of a d dimensional CA with the state set S is a mapping

$$c: ZdS \quad \text{--------} \quad (2)$$

The state of a cell € Zd is c(). The set of all configurations is. Vidushi Sharma et al [3] studied various types of neighborhoods named after the scientist who invented, in Cellular Automata. Some of them are Von Neumann neighborhood, Moore neighborhood, Extended Moore neighborhood and Margolus neighborhood. Fig 1.1 to 1.4 shows these types of neighborhoods.

In Von Neumann neighborhood, the north, west, east and south neighborhood cells are considered for the local function to generate the next generation configuration.

In Moore neighborhood, in addition to the Von Neumann neighborhood cells, it considers the diagonal cells also. In an extended Moore neighborhood, the neighborhood is extended by a distance of two beyond the one. Figure 2 shows the classification of CA.

CA is classified into two types based on the capability of changing cell status on the fly. They are Programmable CA and Controllable CA. Figures 3.1 and 3.2 show these types.

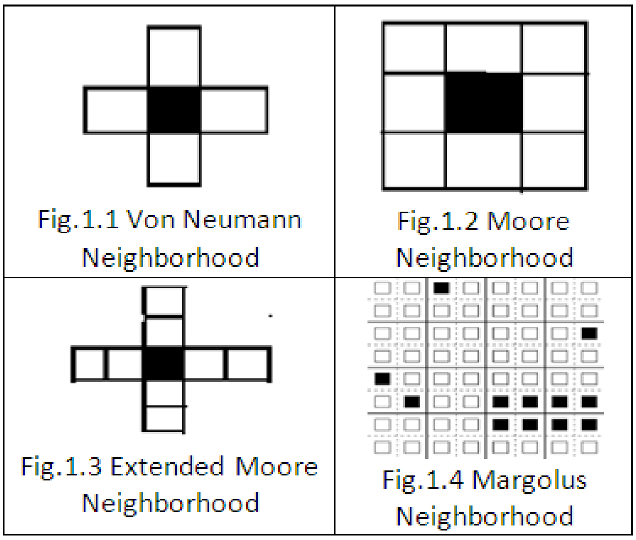


Fig.1.1 Von Neumann Neighborhood

Fig.1.2 Moore Neighborhood

Fig.1.3 Extended Moore Neighborhood

Fig.1.4 Margolus Neighborhood

Figure 1: Types of Neighborhood

A. Programmable CA (PCA)

In general, only one CA configuration could be evolved when rules are applied to an initial configuration at discrete times. But the authors in [4] proposed a novel CA called Programmable CA (PCA) which allows the evolution of multiple CA configurations from the same initial configuration at a discrete-time. A Programmable CA (PCA) is a CA in which the action (how the state of a cell is updated in each cycle) of some cells can be controlled via rule control signals for each cell. [5]. If a cell can change its local rule to be applied at each time step, then it is said to be Programmable CA and it is widely used in VLSI Context. Figure 3.1 shows a single cell structure of a PCA. This PCA allows one control input per cell that configures the rule applied to that cell either to rule 90 or rule 150. If the rule control word contains 1 in the ith bit then rule 150 is applied to the ith cell of the PCA and if rule control word contains 0 in the ith bit then rule 90 is applied to the ith cell of the PCA. For example, the rule control word <0110> for a four-cell PCA allows the first and fourth cell to be configured with rule 90 and second and third cell of the PCA to be configured with rule 150.

B. Controllable CA (CCA)

A Controllable CA (CCA) is a CA in which the action (how the state of a cell is updated in each cycle) of some cells can be controlled via cell control signals in addition to the rule control cells [4]. They reported that the cell which is under the control of a cell control signal is called a controllable cell. If it is not controlled by the cell control signal, then it is called a basic cell. The basic cell acts like the cells of PCA. The CCA is the combination of controllable cells and basic cells.

According to the authors in [4], there are two kinds of CCA cells. They are Programmable Controllable cell and non-programmable controllable cells. The evolution of the next generation depends on the status of the controllable cell. A controllable cell may be either activated or normal. When it is normal, the computation of next state transition is similar to a programmable CA i.e. based on the rule control signal and the states of its neighbors. When it is activated, the computation of the state transition is performed by some set of predefined actions. The programmable control cell is depicted in figure 3.2.

Also CA is classified as two types based on the arrangement of cells. They are one dimensional and two dimensional CA.

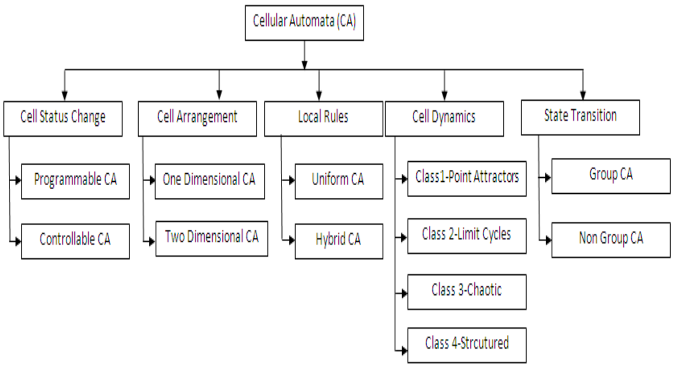


Figure 2: Classification of Cellular Automata

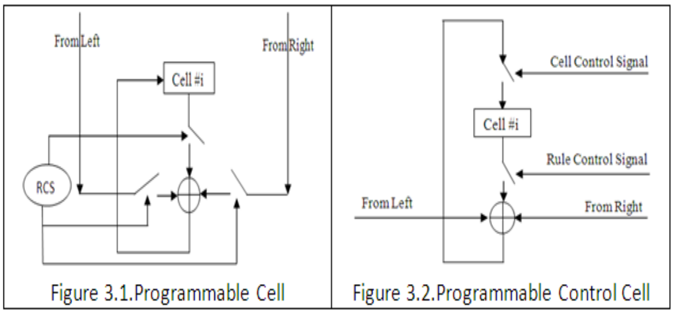


Figure 3.1.Programmable Cell

Figure 3.2.Programmable Control Cell

Figure 3: Structure of a Programmable Cell and Programmable Control Cell

C. One dimensional CA

Stephen Wolfram [7] stated that one dimensional CA consists of a line of sites with values ai between 0 to k-1, where k is the number of permissible states each site in CA can have. The value of ai is updated in discrete time steps according to a deterministic rule depending on the surrounding neighborhood.

$$a_i(t+1)=\phi[a_{i-r}(t), a_{i-r+1}(t), \ldots a_{i+r}(t)] \quad \text{-------- (3)}$$

Where r is the radius of the neighborhood. Each cell will have 2r+1 no of neighbors.

D. Two dimensional CA

Two dimensional CA is a rectangular grid of cells. The neighborhood cell of each cell may be five (Von Neumann's Neighborhood), or nine (Moore neighborhood). A two dimensional CA was proposed by John Conway, called “Game of Life”, which is based on biological Model. Each cell can have two possible states i.e. 1 (alive) or 0 (dead). The state of a cell may depend only on the states of the neighboring cells in the previous time step or it may depend on states of neighbor over several previous times (Memory CA). The local rule is described as follows:

Survival: If a cell is in state 1 (alive) and has 2 or 3 neighbors in state 1, then the cell survives, i.e., remains in state 1 in the next generation. *Birth*: If a cell is in state 0 and has exactly 3 neighbors in state 1, then in the next time step the cell goes to state 1.

Death: A cell in state 1 dies (goes to state 0) of loneliness if it has 0 or 1 live neighbor. Also, it dies because of overcrowding if it has 4 or more live neighbors.

E. Uniform CA

The CA in which the same rule is applied to all the cells to generate the successive generation.

F. Hybrid CA

The CA in which different rules can be applied to the cells to generate the successive generation

CA is said to be dynamic because it evolves as successive time. Wolfram classified CA into four broad classes based on the study of the dynamic properties of CA, [13]. This is shown in figures 4.1 to 4.4.

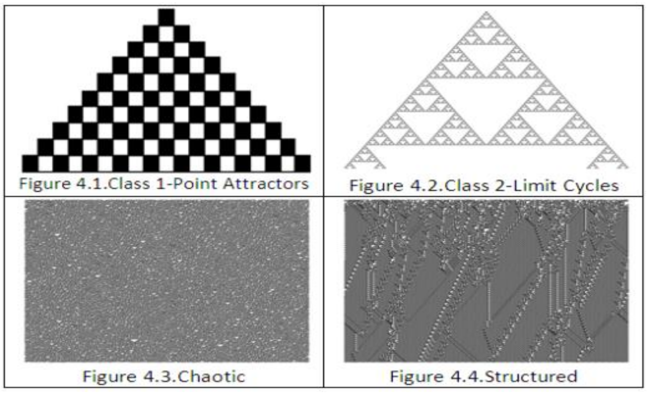


Figure 4.1.Class 1-Point Attractors

Figure 4.2.Class 2-Limit Cycles

Figure 4.3.Chaotic

Figure 4.4.Structured

Figure 4: Wolfram’s Classes of CA

G. Class 1-Point Attractors

CAs in this class 1 will evolves into a homogenous stationary arrangement, with every cell in the same state. E.g) Rule 150 of the Wolfram’s rules for elementary CA.

H. Class 2-Limit cycles

CAs in this class form periodic structures that endlessly cycle through a fixed no of states. E.g) Rule 90 of the Wolfram’s rules for elementary CA.

I. Class 3-Chaotic

CAs in this class form a-periodic random like patterns that resemble white noise on a bad T.V channel, and are sensitive to initial conditions. E.g. Rule 30.

J. Class 4-Structured

CA in this class form complex patterns with localized structures that move through space over time. E.g. Rule 110.

Based on the state transition diagram of the CA, they are classified as two types.

K. Group CA

A CA is referred to as Group CA if all the states in the state transition diagram lie on cycles.

L. Non-Group CA

A CA is referred to as Non-Group CA if all the states in the state transition diagram do not lie on cycles. Dipanwita Roy Chowdhury et al [9] reported the properties of a non-group CA and it’s dual. This type of CA has been proposed as an ideal test machine.

III. Related Works

Stephen Wolfram is the first who applied CA in the field of cryptography where pseudo-random numbers play an important role. In his paper [11], he discussed about one dimensional CA which generates random sequence with a simple rule as

ai(t+1)= ai-1(t) XOR (ai(t) OR ai+1(t)) -------- (4)

These sequences of randomness are analyzed by a variety of empirical, combinatorial, statistical, dynamic systems theory and computation theory methods. The aforementioned rule is applied and a random sequence has been generated. Also he analyzed the possible patterns generated by evolution with this rule starting from all possible initial configurations. Also he considered the changes in the patterns produced by small perturbations in the initial state i.e. the stability properties. It has been noticed that any local changes in the state of any sites will propagate throughout the CA. The patterns evolved from a particular special initial configuration have been analyzed. He used two rules i.e. Rule 30 and Rule 45. It has been reported that rule number 30 which is a non-linear rule seems to be the best as a random sequence generator.

Hortensius et al [12] proposed a pseudo-random number generator based on CA. The random sequence generated here is used for parallel processing. If the conventional pseudo-random number generation technique like congruential methods is employed for this kind of parallel processing, it will require excessive area and more computation time. So CA-based technique is a suitable one. In this article they used binary one dimensional CA where the state of the next site depends only on itself and its closest left and right neighbors. The null boundary condition (i.e. the first and last sites consider their missing neighbor site to always have a zero value) is used. Rule 30 and Rule 150 are applied separately and tested for randomness. This work has been compared with the conventional linear pseudo-random number generator. Marco Tomassini et al [13] presented evolutionary cellular programming which differs slightly from the conventional evolutionary genetic algorithms. They have stated two main differences between the conventional standard genetic algorithm and their evolutionary cellular programming. Firstly the fitness ranking function is evaluated locally at every CA wherein the traditional standard genetic algorithm, a global fitness function is evaluated. Secondly it differs in the way of evolution of the CA rule set. Generally each CA is run independently after which the genetic operators like mutation and cross over are applied. But here the CA coevolves. The diehard test has been conducted for the sequence generated by the evolved rule set. The chi-square test and the correlation factor have been used to analyze the correlation between the sequences. Kolmogorov–Smirnov test has also been conducted.

The aforementioned authors [14] later attempted to generate high-quality random numbers by two dimensional (2D) CA evolved by cellular programming evolutionary approach. Similar to the previous approaches, the good set of rule tables are evolved using this evolutionary cellular programming for two dimensional CA. To generate the next generation, the individuals (rule table–genome) are selected according to the results of fitness evaluation and are transformed by applying genetic operators (mutation and cross over). It has been reported that the 2D CA generated by this evolutionary cellular programming can generate a very good quality random number sequence when compared to one dimensional CA. Also they analyzed the cycle length of the 2D CA generated for different sizes of the CA. It has been reported that when the size of CA increases, then the cycle length also increases. In [15], the authors used nonuniform one and two dimensional CA as a keystream generator. Here the non-uniform rules set have been evolved by evolutionary algorithms. The random mixture of the rules 90,105, 150, 165 has been used. The key K is formed from the current rule vector Ri and the initial random state of CA(0). So for an N bit CA, the apparent length of a key is 4N X 2N= 23N. For the two dimensional CA, instead of 4 rules, seven additive rules are used. The apparent length of a key is 7N X 2N which is larger than the key generated on one dimensional CA. But two dimensional CA is somewhat harder to implement in the hardware. The key could be changed at will so that it is more secure. The randomness of the random sequence can be increased in two-dimensional Cellular Automata compared to one dimensional CA. But the two dimensional CA has more complexity so that it is too difficult to implement in hardware. As discussed earlier, randomness can be increased by evolving the one-dimensional Programmable CA. Sheng-Uei Guan et al [16], proposed a special Programmable Cellular Automata called Self-programmable CA (SPCA). SPCA is more complex when compared to PCA. It has an additional rule selection neighborhood for dynamic rule selection. The rules to be applied are decided dynamically by switching it between the two-state transition rules (150/105 and 90/165) programmed. They used two uniform SPCA with rule combinations SPCA 90/165 and SPCA 150/105. The test (ENT and DIEHARD) results have been analyzed for over 20 sequences averaged and compared with maximum length hybrid 90/150 CA and uniform CA90 and Uniform CA 150 results. It has been proved that SPCA 150/105 and SPCA 90/165 attained the best results. Dan Mocanu et al [17] proposed an SPCA called Global Feedback Self Programmable Cellular Automaton (GF-SPCA). This CA uses its output to update its internal rule set to improve the randomness quality. Instead of local feedback for every cell (in SPCA), it has global feedback in which the internal rules are updated based on the entire CA state not an individual single-cell state. The random number sequences generated by GF-SPCA have been tested against the test suites such as ENT and NIST. Also this GF-SPCA has been implemented in the FPGA and the resource utilization has been investigated and compared with other existing PRNG.

The authors in [18] used an evolutionary algorithm to generate high-quality random numbers by two dimensional CA. They evaluated the quality of the generated sequence by applying a suite of randomness tests. A highly disordered encryption algorithm was proposed in [19] based on two dimensional CA. They presented around 24 rules to be applied in the 2D CA for the generation of PRN. They used the only uniform CA i.e. the same rule is applied to all the cells. They have determined some metrics such as average entropy, hamming distance and Lyapunov exponent.

IV. PROPOSED SYSTEM

A. Two Dimensional Programmable Cellular Automata

Realizing different CA configurations on the same structure can be achieved using a Rule Control Signal (RCS) to control the selection of rules to be applied. Thus different rules can be applied to the same cell at different clock cycles thereby increasing the cycle length. Such a structure is referred to as a Programmable CA (PCA). Only the PCA structure for one dimensional CA has been proposed so far. Here a two-dimensional Programmable Cellular Automata (PCA) with one rule control signal, two rule control signals and three rule control signals are presented. The structure of a hybrid two-dimensional PCAs with one control signal, two control signals and three control signals are shown in figure 5, figure 6and figure 7 respectively.

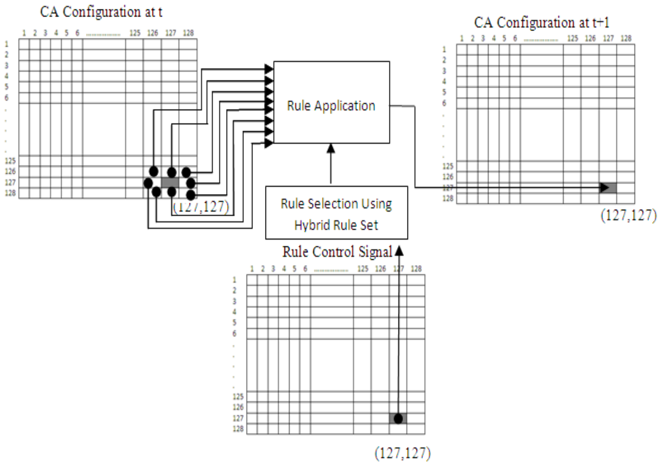


Figure 5: Structure of PCA with one Rule Control Signal

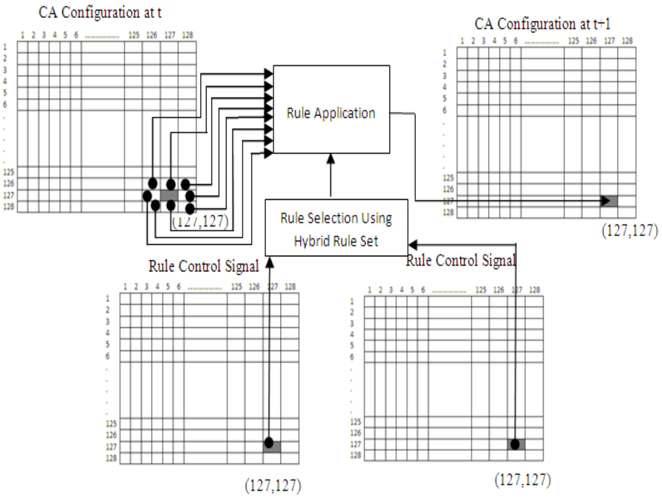


Figure 6: Structure of PCA with two Rule Control Signals

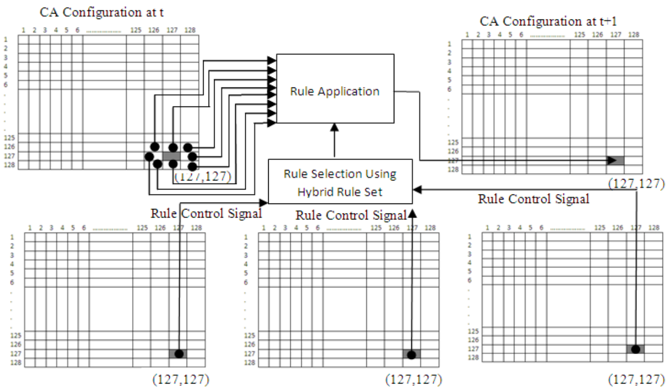


Figure 7: Structure of PCA with three Rule Control Signals

Since the rules applied to the same cells at different clock cycles are different, the sequence generated by this PCA is unpredictable. The PCA used here is 128 X 128 CA and the rule control CA word is also generated by a 128 X 128 CA by applying one of the rules described below. For the evolution of rule control CA, the rule B1357/S2468 is used. Also, the rules involved in the hybrid ruleset are as follows:

B1357/S2468 Rule: If a cell at time t is dead(0) and the total number of living (1) cells around it are one or three or five or seven, then it becomes alive(1) at time t+1. If a cell at time t is alive and the total number of live cells around it is two or four or six or eight, then it retains its state.

B357/S1358 Rule: If a cell at time t is dead(0) and the total number of living (1) cells around it are three or five or seven, then it becomes alive(1) at time t+1. If a cell at time t is alive and the total number of live cells around it is one or three or five or eight, then it retains its state.

B34/S34 Rule: If a cell at time t is dead (i.e. 0) and the total number of life (i.e.1) cells around it are three or four, then it becomes alive (1) at time t+1. If a cell at time t is alive and the total number of live cells around it is one of three or four, then it retains its state.

B3/S1234 Rule: If a cell at time t is dead (i.e. 0) and the total number of life (i.e.1) cells around it are three, then it becomes alive (1) at time t+1. If a cell at time t is alive and the total number of live cells around it is one or two or three or four, then it retains its state.

B3/S12345 Rule: If a cell at time t is dead (i.e. 0) and the total number of life (i.e.1) cells around it are three, then it becomes alive (1) at time t+1. If a cell at time t is alive and the total number of live cells around it is one or two or three or four or five, then it retains its state.

B/S1236 Rule: If a cell at time t is dead (i.e. 0), then it becomes life (i.e.1) at time t+1 without any constraint. If a cell at time t is alive and the total number of live cells around it is three or six, then it retains its state.

B23/S36 Rule: If a cell at time t is dead (i.e. 0) and the total number of living (1) cells around it are two or three, then it becomes alive(1) at time t+1. If a cell at time t is alive and the total number of live cells around it is three or six, then it retains its state.

B45678/S2345 Rule: If a cell at time t is dead (i.e. 0) and the total number of life (i.e.1) cells around it are four or five or six or seven or eight, then it becomes alive(1) at time t+1. If a cell at time t is alive and the total number of live cells around it is two or three or four or five, then it retains its state.

B. *Hybrid Rule Selection*

The hybrid ruleset are framed for PCA with one RCS, two RCS and three RCS are < B1357/S2468, B/S1236>, < B357/S1358, B34/S34, B/S1236, B23/S36> and < B3/S1234, B357/S1358, B34/S34, B/S1236, B23/S36, B45678/S2345, B3/S1234, B3/S12345> respectively. The value of RCS1, RCS2, and RCS3 decides the rule to be applied. For the testing purpose the number of iteration is selected as 100.

C. PCA With One RCS

The Rule Control Signal (RCS) word is also a 128 X 128 CA which is evolved by using rule B1357/S2468. For each time of iteration of the PCA which is involved in the generation of PRN sequence, the newly evolved RCS word is used. The rule set framed for this case is <B1357/S2468, B/S1236>. The state transition of a cell (i,j) at time t depends on the state of the cell (i,j) of RCS. If the cell (i,j) is one, then the rule B1357/S2468 is applied to generate the state of the cell(i,j) at time t+1, otherwise, the rule B/S1236 is applied. The rule to be applied for the cell (i,j) is as follows,

$$\Phi_1^{i,j} = \quad R_{one} \text{ [RCS1 [i, j] \%2] ------ (5)}$$

Where R $_{one}$ is an array of rules {B1357/S2468, B/S1236}

D. PCA with two RCS

The Rule Control Signals (RCS1 and RCS2) words are also a 128 X 128 CA which is evolved by using rule B1357/S2468. For each time of iterations of the PCA involved in the generation of PRN sequence, the newly evolved corresponding RCS word is used. The rule set framed for this case is <B357/S1358, B34/S34, B/S1236, and B23/S36>. The state transition of a cell (i,j) at time t depends on the state of the cell (i,j) of RCS1 and state of the cell (i,j) of RCS2. The rule to be applied for the cell (i,j) is $\Phi_2^{i,j}$ which is obtained by the following equation.

$$\Phi_2^{i,j} = \quad R_{two} \text{ [BD (RCS1 [i, j], RCS2[i, j]) \%4] ------ (6)}$$

Where R $_{two}$ is an array of rules {B1357/S2468, B357/S1358, B23/S36, B/S1236} and BD – a function which converts the sequence of bits to its equivalent decimal?

Table 1: Rule Selection based on Rule Control Signals for PCA with two RCS

RCS1	RCS2	Rule Selected
0	0	B1357/S2468
0	1	B357/S1358
1	0	B23/S36
1	1	B/S1236

E. PCA with three RCS

The Rule Control Signals (RCS1, RCS2, and RCS3) words are also a 128 X 128 CA which is evolved by using the rule B1357/S2468. For each time of iteration of the PCA involved in the generation of PRN sequence, the newly evolved corresponding RCS word is used. The rule set framed for this case is <B3/S1234, B357/S1358, B34/S34, B/S1236,

B23/S36, B45678/S2345, B3/S1234, and B3/S12345>. The state transition of a cell (i,j) at time t depends on the state of the cell (i,j) of RCS1, RCS2, and RCS3. The rule to be applied for the cell (i,j) is $\Phi_3^{i,j}$ which is obtained by the equation (7).

$\Phi_3^{i,j}$ = R_{three} [BD (RCS 1[i, j], RCS 2[i, j], RCS 3[i,j]) %8] ------ (7)

Where R_{three} ={ B3/S1234, B357/S1358, B34/S34, B/S1236, B23/S36, B45678/S2345, B3/S1234, B3/S12345} and BD – a function which converts the sequence of bits to its equivalent decimal.

Table 2: Rule Selection based on Rule Control Signals for PCA with three RCS

RCS1	RCS2	RCS3	Rule Selected
0	0	0	B3/S1234
0	0	1	B357/S1358
0	1	0	B34/S34
0	1	1	B/S1236
1	0	0	B23/S36
1	0	1	B45678/S2345
1	1	0	B3/S1234
1	1	1	B3/S12345

The Pseudocode for PCA with one RCS, two RCS and three RCS are shown in Annexure 1.

V. Performance Analysis

A. Chaotic behavior measure

The following metrics are used as a measure of the chaotic behavior of PCA with RCS applied as the pseudo-random number sequence generated by this proposed PCA.

Entropy[19]:

It is a measure of the order of disorder or randomness in a closed system or universe. Entropy applied to the CA can be determined as

$$H= -\sum_{i=0}^{2^h-1} P_i \log_2 P_i \quad \text{---------- (8)}$$

Where Pi is the probability of the state i and k is the number of possible states. The value of H lies between 0 and 1. The maximum value of H is 1 which means that a high degree of unpredictability. In this work we have calculated average entropy for the given number of clock cycles. h is the length of the sequence of bits tested for entropy. If h=1, then H=P1 log2 P1 + P2 log2 P2. If h=2, then H=P1 log2 P1 + P2 log2 P2 + P3 log2 P3 + P4 log2 P4 i.e. the possible sequences in two-state transition are 00, 01, 10 and 11.

Hamming Distance [19]:

It is a convenient measure of distance, between any two binary CA configurations. It can be defined as the minimum number of substitutions required to change one state ci into another. Also it is the number of differences between states of the cells at time t and t+1. As higher the changes, the lower the repetition.

$$DH= \frac{\sum s(.,t)\ XOR\ s(.,t+1)}{size(\pi)} \quad \text{---------- (9)}$$

The maximum DH is closed to 1. size(π) is the size of two dimensional CA. If the CA is of 3 x 3, then size(π) is 9.

Cycle Length:

It is the maximum number of generations generated from a given initial seed without repetition. If the cycle length is 2n-1, then it is said to be the maximum length where n is the size of CA.

All three architectures i.e. PCA with one RCS, PCA with two RCS, PCA with three RCS yields good entropy, maximum hamming distance, and maximum cycle length which is shown in figure 10. These PCAs are run through 100 clock cycles and sequences are generated individually.

B. Extraction of Pseudo-Random Number sequence

Before feeding the generated sequence to the Diehard tester, the pseudo-random numbers are extracted as follows. The sequence obtained has a sequence of 128 X 128 binary matrices. Every random number to be extracted is of 16 bits in size. To reduce the repetition, the site spacing of two is introduced here. To increase the randomness and reversibility for the application in cryptography, the consecutive bytes are XORed with each other. The consecutive bytes of size eight are XORed with each other and the collection of four such XORed bytes is considered as a random number.

$$\overrightarrow{R_i} = B_i \oplus B_{i+1} \quad \text{------------- (10)}$$

Then this refined sequence is fed into the Diehard test suite and all the tests got passed.

VI. Experimental Analysis

The results presented in the following table 3 and 4 show that, our proposed PCA hybrid rule set with a different number of rule control signals exhibits a better chaotic behavior. Table 3 shows the chaotic behavior of the rules proposed in [19] and the proposed PCAs when the initial seed has taken has high entropy.

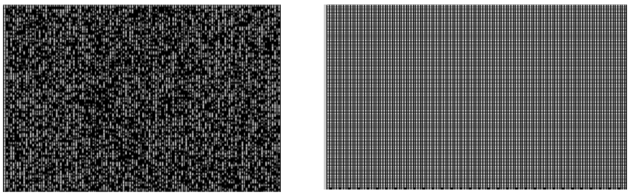
Table 3: Initial Seed with high entropy value (H=1.0)

Rules	Average Entropy			Average Hamming Distance
	h=1	h=2	h=3	
B1357/S2468 [19]	0.9999	1.5036	2.0108	0.5119
B/S1236	0.7200	0.7243	1.3056	0.9612
B357/S1358 [19]	0.9728	1.5467	2.0026	0.4928
B34/S34 [19]	0.9811	1.4287	1.9209	0.5099
B23/S36 [19]	0.9688	1.5992	2.0576	0..5714
B3/S1234	0.9904	1.1735	2.1154	0.1836
B45678/S2345	0.9925	1.1342	1.8008	0.7101
B3/S12345	0.9897	1.4993	2.0872	0.1732
Rule Set of PCA with one RCS	1.0000	1.5064	2.0115	0.5118
Rule Set of PCA with two RCS	0.9746	1.5438	1.9935	0.5026
Rule Set of PCA with three RCS	0.9998	1.5503	2.4420	0.2383

Table 4 shows the degree of chaotic behavior when the initial seed is chosen has low entropy value over 100 clock cycles. These two tables proved that the proposed PCA with Rule control Signal has a high degree of chaotic even when the initial seed chosen has very low entropy.

But this was not the case for the rules in [19]. Also it has been analyzed that even though the ruleset of PCA with three Rule Control Signal has a low hamming distance, it has the best entropy value in all the cases of sequence length h. Table 6 shows that the cycle length is high for the proposed rule set even if the chosen initial seed has very low entropy. Annexure 2 shows the pattern generated at various clock cycles by plotting black dots if the state of that particular cell is 1 and while dot otherwise when the initial seed is chosen has very low entropy. So it is proved that the patterns generated by the proposed PCA with one, two or three RCS are random irrespective of the initial seed chosen. Table 4: Initial Seed with low entropy value (H=0.0204)

Rules	**Average Entropy**			**Average Hamming Distance**
	h=1	**h=2**	**h=3**	
B1357/S2468 [19]	0.6416	1.1.87	0.0000	0.39985
B/S1236	0.0220	0.0000	0.0000	0.0020
B357/S1358 [19]	0.0000	0.0000	0.0000	0.0020
B34/S34 [19]	0.0000	0.0000	0.0000	0.0020
B23/S36 [19]	0.0000	0.0000	0.0000	0.0020
B3/S1234	0.0000	0.0000	0.0000	0.0020
B45678/S2345	0.0000	0.0000	0.0000	0.0020
B3/S12345	0.0000	0.0000	0.0000	0.0020
Rule Set of PCA with one RCS	1.0000	1.5064	2.0115	0.5118
Rule Set of PCA with two RCS	0.9963	1.5036	2.0147	0.5506
Rule Set of PCA with three RCS	0.9628	1.4604	1.9306	0.4250



Initial Seed with entropy H=1

Initial Seed with entropy H=0.0204

Figure 8: Pattern of Initial Seed

Figure 8 shows the pattern of initial seed generated at various clock cycles.

The pseudo-random number sequence is extracted as explained in the previous sections. This sequence extracted from the generated sequence of 128 X 128 by the PCA with a different number of rule control signals is tested against a statistical test called Diehard. Table 5 summarizes these results. Figure 8 shows the entropy value for PCAs with different number of RCS. When the sequence that has to be tested for randomness his 1, it has a very high degree of randomness since the values are very closer to 1. But when the sequence is 2 or 3, the degree of randomness becomes lower as the values are not so closer to 2 and 3 respectively. After the extraction of a sequence of 16-bit random numbers, it is fed to the test suite to test for randomness.

Table 5: Results of the Diehard test

S.No	Test Name	PCA with one RCS	PCA with two RCS	PCA with three RCS
1	Birthday Spacing	Pass	Pass	Pass
2	Binary rank 31X31	Pass	Pass	Pass
3	Binary rank 32X32	Pass	Pass	Pass
4	Binary rank 6X8	Pass	Pass	Pass
5	Minimum Distance	Pass	Pass	Pass
6	DNA	Pass	Pass	Pass
7	Squeeze	Pass	Pass	Pass
8	Runs	Pass	Pass	Pass
9	Count the 1's	Pass	Pass	Pass
10	Count the 1's specified bytes	Pass	Pass	Pass
11	Monte Carlo value for pi	Pass	Pass	Pass
12	Overlapping 20 tuples bit stream	Pass	Pass	Pass
13	Overlapping Pairs Sparse Occupancy	Pass	Pass	Pass
14	Overlapping Quadruples Sparse Occupancy	Pass	Pass	Pass

Table 6: The Maximum cycle length for the PCA with different numbers of RCS when the initial seed selected has high entropy H=1and compared with the rules in [19].

Two dimensional CA Rules	**Obtained Maximum Cycle Length.** **(Theoretical Maximum Cycle Length 2^{16} = 65536)**	
	The initial seed has high entropy H=1.	**The initial seed has low entropy H=0.0204.**
B1357/S2468 [19]	500	2
B357/S1358[19]	1000	2
B34/S34[19]	500	2
B23/S36[19]	250	2
B3/S1234[19]	1000	2
B45678/S2345[19]	100	2
B3/S12345[19]	500	2
PCA with one Rule Control Signal [Proposed]	5000	1000
PCA with two Rule Control Signals [Proposed]	5000	1800
PCA with three Rule Control Signals [Proposed]	6000	500

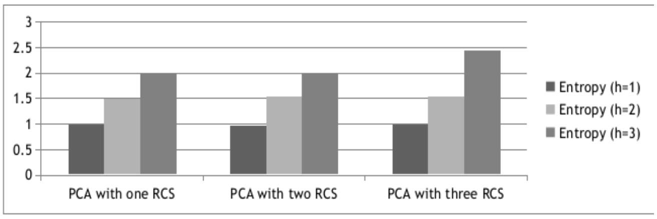


FIGURE 9: ENTROPY VALUE FOR PCA WITH 1, 2 AND 3 RCS WITH H=1,2 AND 3

Figure 10 shows the comparison of metrics such as entropy, hamming distance and cycle length of the obtained random sequence for the proposed PCA with different numbers of RCS.

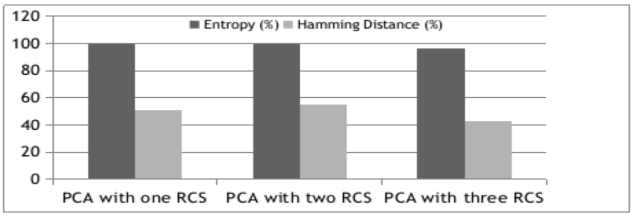


Figure 10: Comparison among PCA with different number of RCS

VII. CONCLUSION

All the three proposed architecture of the PCA hybrid ruleset acts as a very good pseudo-random number generator in terms of chaotic measures like entropy, hamming distance, and cycle length. They also satisfy the statistical tests involved in the Diehard test suite, since they achieve higher randomness quality irrespective of the randomness of the initial seed chosen. The proposed work will be extended by encrypting images with random sequences generated by the proposed pseudo-random number generator using PCA with a different number of rule control signals and checking the performance of the encryption process.

REFERENCES

1. Sandip Karmakar, Debdeep Mukhopadhyay and Dipanwita Roy Chowdhury. (2010) “d-Monomial Tests of Nonlinear Cellular Automata for Cryptographic Design”, 9th International Conference on Cellular Automata for Research and Industry, ACRI 2010, Italy, , pp 261-270.
2. Harald Niesche, “Introduction to Cellular Automata”. (2006), Seminar on Organic Computing.
3. Vidushi Sharma, Anurag Dev and Sachin Rai. (2012) .“A comprehensive study of Cellular Automata”. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, pp.340-344.
4. Sukumar Nandi, Pushkar Pal and Parimal Pal Chaudhuri. (1994). “Theory and Applications of Cellular Automata in Cryptography”. IEEE Transactions on Computers, Vol.43, pp. 1346-1357.
5. Sheng-Uei Guan and Shu Zhang. (2002).“A Family of Controllable Cellular Automata for Pseudorandom Number Generation”. International Journal of Modern Physics C, Vol. 13, pp.1047-1073,.
6. Taejon Chang, Lickho Song, Jinsoo Bae and Kwang Soon Kim. (1997). “Maximum Length Cellular Automaton sequences and its application”. Journal of Signal Processing, Vol.56, pp.199-203.
7. Stephen Wolfram. (1985). “Origin of randomness in physical systems”, Journal of Physics Review Letters, Vol.55, pp. 449-452.
8. Stephen Wolfram. (1984). “Cellular automata as models of complexity”, Journal of Nature, Vol. 311, pp. 419-424.
9. Stephen Wolfram. (1984). “Universality and Complexity in Cellular Automata”. Journal of Physica D – Nonlinear Phenomena, Vol.10, Issue 1, pp.1-35.
10. Supratik Chakraborty, Dipanwita Roy Chowdhury, and Parimal Pal Chaudhuri. (1996). “Theory and Application of Nongroup Cellular Automata for Synthesis of Easily Testable Finite State Machines”, IEEE Transactions on Computers, Vol.45, pp. 769 - 781.
11. Kaushik Chakraborty and Dipanwita Roy Chowdhury. (2012).“CSHR: Selection of Cryptographically Suitable Hybrid Cellular Automata”, 10th International Conference on Cellular Automata for research and Industry, ACRI 2012.
12. Stephen Wolfram. (1986). “Random Sequence Generation by Cellular Automata”. Journal of Advances in Applied Mathematics, Vol.7, pp.123-169.
13. Hortensius, McLeod and Card. (1989). “Parallel Random Number Generation for VLSI Systems Using Cellular Automata”. IEEE Transactions on Computers, Vol.38, pp. 1466 - 1473.
14. Marco Tomassini, Moshe Sipper, Mose Zolla and Mathieu Perrenoud. (1999). “Generating high-quality random numbers in parallel by cellular automata”. Journal of Future Generation Computer systems, Vol.16, pp.291-305.
15. Roy Chowdhury, Sengupta and Pal Chaudhuri. (1994). “A Class of Two-Dimensional Cellular Automata and Applications in Random Pattern Testing”. Journal of Electronic Testing. Vol.5, pp.67-82,.
16. Marco Tomassini and Mathieu Perrenoud. (2001). “Cryptography with cellular automata”. Journal of Applied Soft Computing, Vol.1, pp.151–160.
17. Sheng-Uei Guan and Syn Kiat Tan. (2004) “Pseudorandom Number Generation with Self-Programmable Cellular Automata”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.23, pp. 1095–1101.
18. Dan Mocanu, Alexandru Gheolbanoiu, Radu Hobincu and Lucian Petrica.(2016). “Global Feedback Self –Programmable Cellular Automaton Random Number Generator”. Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia, Vol.39, pp.1-9.
19. Marco Tomassini and Moshe Sipper. (2000). “On the Generation of High-Quality Random Numbers by Two-Dimensional Cellular Automata”. IEEE Transactions on Computers, Vol.49, pp. 1146-1151.
20. Janeth Machicao, Anderson Marco and Odemir Martinez Bruno. (2012). “Chaotic encryption method based on life-like cellular automata”. Journal of Expert Systems with Applications, Vol.39, pp.12626–12635.
21. Wentian Li, Norman Packard and Chris Langton. (1990). “Transition Phenomena in Cellular Automata Rule Space”. Journal of Physica D-Nonlinear Phenomena, Volume 45, pp.77-94.

AUTHORS PROFILE

P Dinakaran received his BE degree from Government College of Engineering in Dharmapuri, India in 2017 and he currently pursuing M.Tech degree in National Engineering College, kovilpatti, in India.

I Gethzi Ahila Poornima received her B.E degree from Manonmanium Sundaranar University, Tirunelveli, India in 2004, M.Tech degree from Kalasalingam University, Krishnankoil, India. She is currently pursuing PhD under Anna University, Chennai. She has worked as Junior Research fellow for DRDO. Her current research interest are in the field of Wireless Sensor Networks, network security and cellular automata.

Dr B.Paramasivan received his BE degree from Madurai Kamaraj University, Madurai, India in 1988, ME degree from Jadavpur University, Calcutta, India in 1994, and PhD from Anna University, Chennai, India in 2009. He is currently working as a professor and head in the Department of Computer Science and Engineering, National Engineering College, Kovilpatti, India. His research interests are in quality of service for wireless networks. Dr. Paramasivan is a reviewer of IEEE Sensors, Computing and Informatics, and Computer Networks and Communications. He has served as a TPC member at various conferences, including the IEEE International Conference on Wireless and Optical Communications. He is a senior member of IEEE, a lifetime member of CSI Mumbai, a member of ISTE New Delhi, and a fellow of the Institution of Engineers (IE), Kolkatta, India.