An Efficient Design of Non-linear CA Based PRPG for VLSI Circuit Testing

Sukanta Das sukd@becs.ac.in
Debdas Dey debdas_dey@rediffmail.com
Subhayan Sen subhayan@becs.ac.in
Biplab K Sikdar biplab@becs.ac.in
P Pal Chaudhuri ppc@becs.ac.in

Computer Science & Technology, B. E. College (a Deemed University), Howrah, India 711103

Abstract— This paper reports the efficient design of Pseudo-Random Pattern Generator ($PRPG$) with linear time complexity. The PRPG is developed around the regular structure of non-linear Cellular Automata (CA). The application of proposed $PRPG$ is demonstrated in designing on-chip Test Pattern Generator (TPG) for VLSI circuits. The quality of the TPG is as good as that designed with the existing schemes, employing maximal length linear CA incurring $O(n^3)$ complexity.

I. Introduction

The $PRPGs$ are traditionally implemented with $LFSR$ or maximal length linear CA [2]. It has been established that the maximal length CA is the better choice for $PRPG$ [1, 2]. However, such a design suffers from the following drawbacks: (i) To generate an $n-$cell maximal length CA we need to find an $n-$degree primitive polynomial. This operation involves exponential complexity. (ii) The design complexity of an $n-$cell maximal length CA from the primitive polynomial is also $O(n^3)$ [4]. (iii) Further, the list of $n-$degree primitive polynomial is available only for $n \leq 500$ [3].

In this background, the major contributions of this paper are: (i) A scheme has been proposed for synthesis of an $n-$cell CA based $PRPG$ with $O(n)$ time complexity. (ii) An analytical framework has been provided to characterize non-linear CA rules generating a $PRPG$ of excellent quality. (iii) The quality of randomness of the proposed non-linear CA based $PRPG$, is as good as that of maximal length linear CA based $PRPG$ involving $O(n^3)$ complexity.

II. Cellular Automata

For a 3-neighborhood 1-dimensional CA, each cell having two states - 0 or 1. The next state of cell i is

$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t)$.

The S_{i-1}^t, S_i^t and S_{i+1}^t are the present states of the left neighbor, self and right neighbor of the i^{th} cell at time t. f_i is the next state function. If f_i is expressed in the form of a truth table, then the decimal equivalent of its output is denoted as the rule $\mathcal{R}_i$. Three such rules 90, 150, and 75 are illustrated in *Table I*. The rules 90 and 150 employ XOR logic. These are linear rules [1]. On the other hand, rule 75 is a non-linear one.

The set of rules that configure the cells of a CA is the rule vector $\mathcal{R} =< \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_i, \cdots, \mathcal{R}_n >$. If all $\mathcal{R}_i$ $(i = 1, 2, \cdots, n)$ are linear, the CA is a Linear CA, otherwise it is a Non-Linear CA. Characterization of linear CA is reported in [1]. A new analytical framework to characterize non-linear CA rules is reported next.

TABLE I
Truth Table for Rule 90, 150 and 75

Present state : (RMT)	111 (7)	110 (6)	101 (5)	100 (4)	011 (3)	010 (2)	001 (1)	000 (0)	$Rule$
(i) Next State :	0	1	0	1	1	0	1	0	90
(ii) Next State :	1	0	0	1	0	1	1	0	150
(iii) Next State :	0	1	0	0	1	0	1	1	75

Definition 1 *A rule is* **Balanced** *if it contains equal number of 1s and 0s in its* $8-$*bit binary representation; otherwise it is* **Unbalanced**.

The rules shown in *Table I* are the balanced rule. On the other hand, rule 171 (10101011) with five 1's is unbalanced.

From the view point of *Switching Theory*, a combination of the present states (as noted in 1^{st} row of *Table I*) can be viewed as *Min Term* of a 3-variable $(S_{i-1}^t, S_i^t, S_{i+1}^t)$ switching function. Therefore, each column of the first row of *Table I* is referred to as **Rule Min Term (RMT)**. Column 011 in the truth table (*Table I*) is the RMT 3. The next states corresponding to this RMT is 1 for rule 90 & 75, and 0 for rule 150.

The state transition diagram of a group CA contains only cyclic states, whereas a non-group CA contains both cyclic and non-cyclic states. If all the states except all 0s state (for linear CA) lie in a single cycle, then it is a *maximal length CA*; otherwise, the group CA is a *non-maximal length* CA [1].

The synthesis of maximal length CA, with $O(n^3)$ complexity, has been reported in [4]. We propose a $PRPG$ synthesis scheme employing non-linear group CA in $O(n)$ time.

III. Synthesis of PRPG with Group CA

Local Randomness in CA cell: A CA cell shows local randomness in its states if the states (0 and 1) are equally probable. Following properties guide the design.

Property 1: A CA cell with a balanced rule displays good quality of randomness compared to an unbalanced one.

Property 2: The next state values of a rule corresponding to each RMT of the pairs (0 & 1), (2 & 3), (4 & 5) and (6 & 7) should be different. Rule 90 & 150 (*Table I*) obey *Property 2*.

Property 3: The next state corresponding to each RMT of the pairs (0 & 4), (1 & 5), (2 & 6) and (3 & 7) of a CA rule should be different. Rule 90 & 150 also obeys *Property 3*.

Property 4: A linear CA configured with 90/150, generating large cycle, displays good quality of pseudo-randomness.

Further, a larger length cycle (of order 2^n) ensures that the probability of occurrence of a global state of an $n-$cell CA

approaches the value $\frac{1}{2^n}$. So the best possible choice for a $PRPG$ is the maximal length CA.

Global Randomness of a CA: The earlier discussions settle design issues for the $PRPG$ displaying global randomness.

C_1: CA rules should be balanced.

C_2: CA rules should satisfy *Property 2* and *Property 3*.

C_3: The CA should have sufficiently large cycle length.

For a large n, an n-cell CA having large cycle length (2^{15}) and its rules satisfying criteria C_1 and C_2 suffice to maintain high quality of pseudo-randomness. The next subsection concentrates on identification of such non-linear Group CA.

A. Non-linear Group CA

Definition 2 *A rule is a* **Non Group Rule** *if its presence in a rule vector makes the CA non-group. A rule is* **Group Rule** *if it is not a non-group rule.*

Theorem 1 An unbalanced rule is a non-group rule.

Example 1 *The $4-cell$ CA with rule vector $< 105, 177, 170, 75 >$ is a group CA. Therefore, all the four are group rules. However, $< 105, 177, 171, 75 >$ is a non-group CA. The presence of rule 171 (10101011) makes the CA non-group. That is, 171 is a non-group rule. The rule 171 is an unbalanced one. The number of 1's in 171 is 5.*

There are ${}^8C_4 = 70$ balanced CA rules in 3-neighborhood. However, out of these 70, only 62 are the group rules. To design a group CA, all the rules in the rule vector should necessarily be group rules. However, the sequence of group rules does not imply that the CA is a group CA.

Example 2 *The CA with rule vector $< 105, 170, 177, 75 >$ is a non-group CA even though all the rules are group rules.*

Example 3 *The $CA < 105, 177, 170, 75 >$ and $< 105, 177, 153, 204 >$ are group CA. But $< 105, 177, 177, \mathcal{R}_4 >$ is non-group, where $\mathcal{R}_4$ is any rule. Rule 177 after the sequence $< 105, 177 >$ leads to the CA as non-group. Therefore, rule $\mathcal{R}_2$ =177 of $< 105, 177, \mathcal{R}_3, \cdots, \mathcal{R}_n >$ allows only a specific set of rules as $\mathcal{R}_3$ for group CA.*

Building Rule Sequence of Group CA: To construct a group $CA < \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_i, \mathcal{R}_{i+1}, \cdots, \mathcal{R}_n >$, we concentrate on a pair of rules $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$. Let S denotes the set of all CA rules. Corresponding to each $\mathcal{R}_i$, there exists a subset of rules $S_i \in S$ such that if $\mathcal{R}_{i+1} \in S_i$, then the CA remains group. The $S_i \in S$ corresponding to a $\mathcal{R}_i$ forms a *class.* It is found that there is only six such classes [5]. *Table II* dictates the class of $\mathcal{R}_{i+1}$ from the class of i^{th} cell and $\mathcal{R}_i$. This consideration enables selection of a rule $\mathcal{R}_i$ ($i = 2, 3, \cdots, n-1$) in the rule vector. The rules $\mathcal{R}_1$ and $\mathcal{R}_n$ (first and the last rules) are selected as follows.

First and Last Rule Tables: There are $2^{2^2} = 16$ effective rules for left most ($\mathcal{R}_1$) and right most ($\mathcal{R}_n$) cells. The RMTs 4, 5, 6 and 7 are as the *don't care* for $\mathcal{R}_1$. So there are only 4 (0, 1, 2, 3) effective RMTs for $\mathcal{R}_1$. Similarly, the effective RMTs for $\mathcal{R}_n$ are 0, 2, 4 & 6.

Corollary 1 *If $< \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n >$ is a group CA, then $\mathcal{R}_1$ and $\mathcal{R}_n$ must be balanced over their effective 4 RMTs.*

TABLE II
CLASS RELATIONSHIP OF $\mathcal{R}_i$ AND $\mathcal{R}_{i+1}$

Class of $\mathcal{R}_i$	$\mathcal{R}_i$	Class of $\mathcal{R}_{i+1}$
I	51, 60, 195, 204	I
	85, 90, 165, 170	II
	102, 105, 150, 153	III
	53, 58, 83, 92, 163, 172, 197, 202	IV
	54, 57, 99, 108, 147, 156, 198,201	V
	86, 89, 101, 106, 149, 154, 166, 169	VI
II	15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240	I
III	15, 51, 204, 240	I
	85, 105, 150, 170	II
	90, 102, 153, 165	III
	23, 43, 77, 113, 142, 178, 212, 232	IV
	27, 39, 78, 114, 141, 177, 216, 228	V
	89, 101, 149, 166, 169	VI
IV	60, 195	I
	90, 165	IV
	105, 150	V
V	51, 204	I
	85, 170	II
	102, 153	III
	86, 89, 90, 101, 105, 106, 149, 150, 154, 165,166, 169	VI
VI	15, 240	I
	105, 150	IV
	90, 165	V

TABLE III
FIRST RULE TABLE

Rules for $\mathcal{R}_1$	Class of $\mathcal{R}_2$
3, 12	I
5, 10	II
6, 9	III

TABLE IV
LAST RULE TABLE

Rule class for $\mathcal{R}_n$	Rule set for $\mathcal{R}_n$
I	17, 20, 65, 68
II	5, 20, 65, 80
III	17, 20, 65, 68
IV	20, 65
V	17, 68
VI	5, 80

There are ${}^4C_2 = 6$ rules, out of 16, that are balanced over their effective 4 RMTs. *Table III* identifies the $\mathcal{R}_1$s and the corresponding class for the rule $\mathcal{R}_2$. Similarly, the balanced rules for rule $\mathcal{R}_n$ are enlisted in *Table IV*.

Example 4 : Synthesis of 4-cell group CA*: Consider, rule 9 is selected as $\mathcal{R}_1$. Therefore, the class (obtained from* Table III*) of $\mathcal{R}_2$ is III. From* Table II *rule 177 is selected randomly as $\mathcal{R}_2$. The class of $\mathcal{R}_3$ is then V. We select 170 as $\mathcal{R}_3$ from* Table II. *The class of last cell is, therefore, II (from* Table II*). Rule 65 is selected randomly for $\mathcal{R}_4$ from* Table IV. *Therefore, the group CA is $< 9, 177, 170, 65 >$.*

B. The PRPG

The algorithm to synthesize the $PRPG$ is next presented.

Algorithm 1 PRPG-Synthesis
Input*: n (CA size),* Tables *II, III* and *IV.*
Output*: A $PRPG$.*
Step 1: *Pick up the first rule $\mathcal{R}_1$ randomly from* Table III *that holds* Property 2 *and set the class of $\mathcal{R}_2$.*
$C :=$ *Class of $\mathcal{R}_2$ ($C \in \{$I, II, III$\}$ of* Table III*).*
Step 2: *For $i := 2$ to $n-1$ repeat step 3 and step 4.*
Step 3: *From* Table II *pick up $\mathcal{R}_i$ that obeys* Properties 2 & 3 *for C.*
Step 4: *Find class C for next cell rule using* Table II.
Step 5: *Pick up $\mathcal{R}_n$ (*Table IV*) that obeys* Property 3 *for class C.*
Step 6: *Form the rule vector.*

The complexity of the algorithm is clearly $O(n)$.

Randomness Quality of PRPG: The randomness property of the synthesized $PRPG$, for different values of n, are studied based on the metric proposed in *DiehardC*. Each test contains a set of $'p'$ values. A test succeeds if the p values are in between 0.025 and 0.975. The result of a particular experimentation, for

TABLE V
RANDOMNESS TEST I

Name of Test	Maxlength CA		Proposed PRPG	
	p val	status	p val	status
Overlapping sum	.261047	success	.718341	success
Runs	1.0000	failure	.976141	success
3D Spheres	.237987	success	.863510	success
Parking Lot	.199333	success	.325202	success
Birthday Spac	.170012	success	.183236	success
Count-the-1's	1.0000	failure	1.0000	failure
Binary Rank 6x8	1.0000	failure	1.0000	failure
Binary Rank 31 & 32	.624139	success	.613167	success
Count-the-1's in Byte	.650000	success	1.0000	failure
Bitstream	.900000	success	.980250	success
Craps	1.0000	failure	.956771	success
Minimum Distance	.105987	success	.433171	success
Overlapping Permu	1.0000	failure	1.0000	failure
DNA	1.0000	failure	1.0000	failure
The Squeeze	1.0000	failure	1.0000	failure

$n = 63$, is shown in *Table V*. The results of extensive experimentation establish that the randomness quality of the patterns generated by non-linear CA based $PRPG$ is as good as that of maximal length linear CA. Such a $PRPG$ is employed as Test Pattern Generator (TPG). The fault efficiency of the TPG is tested on a large number of *ISCAS benchmark* circuits.

Table VI compares the fault coverage figures shown by the maximal length linear CA (Column 4) and the proposed TPG (Column 5). The FFs of sequential circuits are assumed to be initialized to 0. It can be observed that in most of the cases (marked with *) the fault coverage of the proposed TPG is same or better than the results obtained with maxlength CA.

IV. CONCLUSION

This paper introduces the concept of non-linear CA in designing the $PRPG$. The design scheme requires $O(n)$ time.

REFERENCES

[1] P Pal Chaudhuri, D.Roy Choudhury, S. Nandi and S. Chattopadhyay, *'Additive Cellular Automata Theory and Applications'*, IEEE Computer Society Press, USA,

[2] P. D. Hortensius, R. D. Mcleod, Werner Pries, D. Michael Miller and H. C. Card, *'Cellular Automata Based Pseudo-random Number Generators for Built-in Self-test'*, IEEE TCAD, Vol. 8, No. 8, 1989.

[3] http://csr.csc.uvic.ca/˜ mserra/CA.html

[4] Kevin Cattel, Jon C. Muzio, *'Synthesis of One-Dimensional Linear Hybrid Cellular Automata'*, IEEE TCAD, vol. 15, no. 3,March 1996.

[5] Sukanta Das, Anirban Kundu, Subhayan Sen, Biplab K Sikdar, P Pal Chaudhuri, *'Non-Linear Cellular Automata Based PRPG Design (Without Prohibited Pattern Set) In Linear Time Complexity'*, ATS, 2003.

TABLE VI
COMPARISON OF TEST RESULTS

Circuit Name	# PI	# Test Vector	Fault Coverage (%)	
			Max Len	TPG
s1196	14	12000	94.85	95.25*
s1238	14	10000	89.67	90.04*
s967	16	9000	98.22	98.22*
s1423	17	15000	56.50	51.16
s1269	18	1200	99.18	99.56*
s3271	26	10000	98.99	98.99*
c6288	32	60	99.51	99.46
c1908	33	4000	99.41	99.41*
s5378	35	8000	67.63	68.06*
s641	35	2000	85.63	85.65*
s713	35	2000	81.41	81.24
s35932	35	14000	61.91	79.95*
c432	36	400	98.67	99.24*
c432m	36	4000	83.57	84.68*
c499	41	600	98.95	98.81
c499m	41	2000	97.78	97.56
c1355	41	1500	98.98	99.24*
c1355m	41	12000	92.23	92.23*
s3384	43	8000	91.78	91.95*
s4863	49	8000	91.83	94.29*
c3540	50	3500	95.85	95.85*
c880	60	2500	99.47	99.05
s991	65	6000	95.06	94.95
s6669	83	4500	99.97	99.99*
c7552	207	12000	94.25	94.44*
c2670	233	2000	84.60	84.57