International Journal of Research Studies in Science, Engineering and Technology
Volume 2, Issue 9, September 2015, PP 72-76
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Hybrid Cellular Automata-Based Pseudo Random Sequence Generator for BIST Implementation

A.Madhuri

Griet, India
amadhuri463@gmail.com

Abstract: *The technique of Test pattern generation plays a key role in Built-In-Self-Test(BIST) architecture implementation. Main problem with any test pattern generator is to produce extended, random path succession which is applied to Circuit Under Test(CUT) for detecting faults. As real long numbers can be acquired as a part of physical developments only, these are complex to be employed inactual applications. Therefore, pseudorandom numbers are generated by artificial design patterns. In this paper we present a variation on a built-in-self-test (BIST) technique, which is based upon a pseudo random number generator derived from a one-dimensional linear hybrid cellular automata (LHCA) array. Certain types of circuit faults are undetectable using the correlated bit streams produced by linear-feedback-shift-register (LFSR). In addition it is noted that CA implementations exhibit data compression properties similar to the LFSR. It is also possible that some of the analysis of pseudorandom testing may be more directly applicable to LHCA-based pseudorandom testing than to LFSR-based schemes. In the last two decades, researchers devoted many efforts to reduce the average power consumption in VLSI systems during normal operation mode, while power consumption during test operation mode was usually neglected. However, during test application circuits are subject to an activity level higher than the normal one: the extra power consumption due to test application may thus rise severe hazards to the circuit reliability. Moreover, it can dramatically shorten battery life when periodic testing of battery-powered systems is considered. In this paper we propose an algorithm to design a Test Pattern Generator based on Hybrid Cellular Automata for testing combinational circuits that effectively reduces power consumption.*

Keywords: *LFSR; LCA; TPG; BIST.*

1. Introduction

Built-In Self-Test being a technique which allows the circuit to test itself without any external equipment has become one of the prominent feature of the chips (especially ULSI,VLSI) being produced. BIST Implementation requires predominantly two components: A Test Pattern Generator and an output Response Analyzer[2]. These components are mostly realized using either Cellular

Automatons or Linear Feedback Shift Registers.An LFSR is a shift register which when clocked,approaches the signal through the register from one bit to the next most-significant bit. Some ofthe outputs are mixed in exclusive-OR configuration to form a feedback mechanism[5]. A cellular Automaton is an infinite horizontal array of square cells, built upon the scheme of nearest neighbor. This paper presents lucid description about a Linear Hybrid Cellular Automata(LHCA), LFSR implementations in BIST in terms of Area, Power. The section wise elucidation about BIST implementation, HYBRID CA, LFSR also their roles as Pseudo Random Pattern Generators has been presented in this paper.

2. BIST-Architecture & Implementation

Built-in self-test is the ability of a circuit (chip, board, or system) to test itself [3]. BIST represents a merger of the concepts of built-in test (BIT) and self-test, and has come to be synonymous with these terms[2]. In order to implement BIST, extra circuitry is added to an existing circuit to enable it to test it-self. The circuitry for BIST consists of four modules: Test Pattern Generator (TPG), Input Isolation Circuitry, Test Controller and Output Response Analyzer (ORA) [4]. The TPG generates pseudo-random test patterns to test the Circuit under Test (CUT). Input isolation circuit isolates the inputs of the circuit during test mode and applies the TPG output to the input of the CUT. The test response of the CUT is analyzed by the ORA. A test controller is provided to sequence and run all the BIST operations. The following figure shows the implementation of BIST.

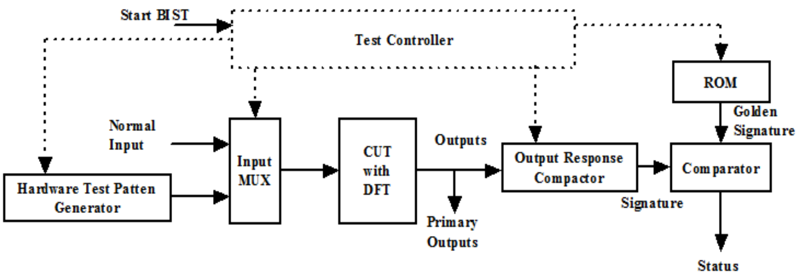


Figure1. *BIST Implementation*

3. Cellular Automata

The need for perfect randomness in the test patterns generated for the purpose of testing the DUT in BIST architecture has been evident through many theories published in the field of "Design for Fault Tolerance and Testability". It can be reached for a larger point through the Cellular Automaton programming explained by Stephen Wolfram in[1] This New kind of Science follows a basic principle which states that "All the complex structures /machines/outputs which are supposed to be the result of more complex construction is not acceptable, as in most of the cases simple programs based on very simple rules are capable of creating the more complex behavior."[1]. However the principle has the effect on many of the fields like viz., Social sciences, Art, Philosophy, Cybernetics, Evolution Theory etc., in this paper the focus have been laid on the field of computation and more specifically for generating the test patterns which would enable us to test the DUT(Device Under Test) in an efficient way. Let us now consider the one-dimensional Cellular Automaton which usually exists on an infinite horizontal array with square cells to be limited to only two states per cell: white and black. In this paper we have considered the CAs whose rules are based on the nearest neighbor scheme. The rule in general states that: "To determine the condition of a cell in position 'q' at time level (n+1), we look at the states of the cells in position (q-1), q, (q+1) all in time step n." [9]. For each of the eight possible patterns of the white and black cells, the state of the cell 'q' at time step (n+1) is chosen as either black or white. Through this there can be eight possible input patterns, as well as onepossible output. In all, we get 256 different possible outputs. Out of 256 different possible outputs, the focus on this paper is laid only on Rule 90 and Rule 150 of CA, because Rule 90 and Rule 150 generates apparent randomness despite the lack of anything that could reasonably be considered random input. Stephen Wolfram proposed using its center column as a pseudorandom number generator (PRNG); it passes many standard tests for randomness, and Wolfram uses this rule in the 'Mathematical' product for creating random integers.

Stephan Wolfram's pioneering research and mathematical analysis of cellular automata led to his famous classification of 1D cellular automaton.

Class 1: The first class of cellular automata always evolves after a finite number of steps from almostall initial states to a homogeneous state where every cell is in the same state. This is something likefixed point equilibrium in the dynamical system.

Class 2: Periodic structures with a fixed number of states occur in the second class of cellular automata.

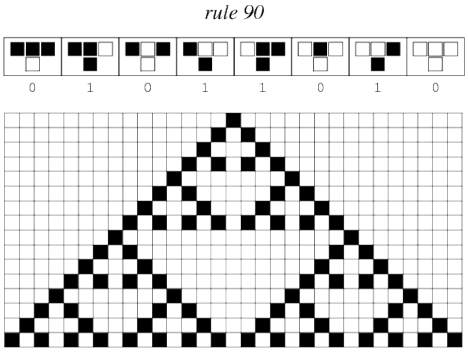
Class 3: Aperiodic or "chaotic" structures appear from almost all possible initial states in this type ofcellular automata.

Class 4: Complex patterns with localized spatial structure propagate in the space as time evolves.

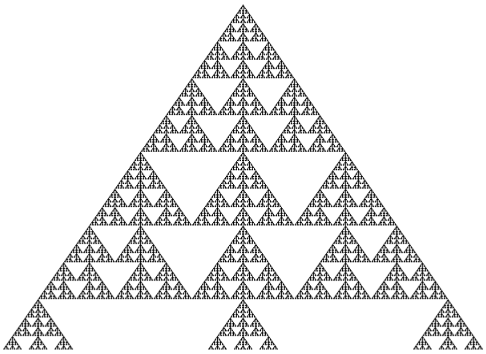
Eventually, these patterns will evolve to either homogeneous or periodic. It is suggested that thisclass of cellular automata may be capable of universal computation.

Cellular automata can be formulated in higher dimensions such as 2D and 3D. One of the most popular and yet very interesting 2D cellular automata using relatively simple updating rules is the Conway's Game of Life. Each cell has only two states $k = 2$ and the states can be 0 and 1.

3.1. Rule 90 of Cellular Automata: Rule 90 states that, if either of the right or left neighbor are black, in the previous step, then takes the new color of the cell to be black otherwise, takes the new color to be white[1].



(b)

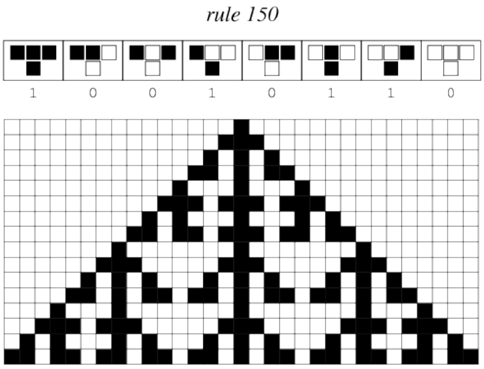


(c)

[1]**Figure2.** *Random patterns generated by Rule 90 of Cellular Automata.*

(a) Pictorial representation of rule 90 with Black cell-1 &white cell-0, Pattern generated after 20 iterations of Rule 90. (b) Pattern generated by 100 iterations of Rule 90.

3.2. Rule 150 of Cellular Automata:Rule 150 states that if there are odd number of blacks along with the cell and its neighbor, then the new state is black otherwise, the opposite color.



[1]**Figure3.** *Random patterns generated by Rule 150 of Cellular Automata.*

(a) Pictorial representation of rule 150 with Black cell-1 &white cell-0, Pattern generated after 20 iterations of Rule 150. (b) Pattern generated by 100 iterations of Rule 150.

Rule 90 and Rule 150 falls under the Class III from random initial conditions. The C.A.'s in this class have shapes that repeat themselves, but their location and frequency is random [1]. This class contains about 4% of the basic CAs.

4. HYBRID CELLULAR AUTOMATA

When next state of the cell is determined using combination of rules in 1-dimensional CA then it is called Linear Hybrid Cellular automata. In, this paper we illustrate hybrid cellular automata test patterns using Rule 90 and Rule 150 as a combination. In the below figure '0' in the cell represents rule 90 and '1' represents Rule 150 [9].

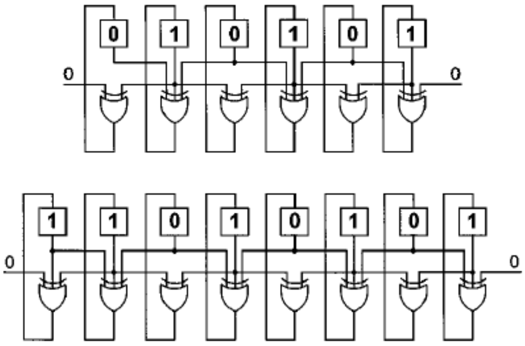


Figure 4. *Example of Hybrid Cellular Automata*

Boundary Conditions:

When extreme right and left cells are considered, to evolve next state of the extreme cells then for those left and right neighbors are not present respectively. To overcome, this problem boundary conditions are evolved[7].

Null boundary conditions: In this to extreme cells '0' is assumed as its neighbor and next state of the cell is evolved according to the rule applied.

Figure 5. *Null boundary conditions representation*

Cyclic boundary conditions: In this to extreme right cell, the extreme left neighbor acts as its neighbor and vice versa happens for extreme left cell.

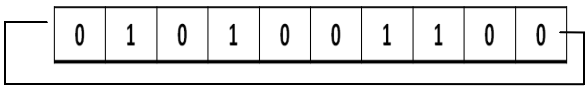


Figure6. *Cyclic boundary conditions representation*

When patterns generated by using null boundary conditions and 1-d LHCA are applied to two different Device Under Test(DUT) which is C3540(ALU) and C432 detection of faults is carried out.

5. Results and Discussion

It briefs differentiating parameters and the results obtained. Power and Area parameters have been considered to differentiate the two TPGs.

It can be clearly seen from the Table1 that CA performs well when it comes to power but gets When it comes to the LFSR, the power consumption and area occupied is peak when compared to

Table 1. *Comparative results based on parameters Power for two TPGs.*

Total Power analysis of Cellular Automata and Linear Feedback Shift Register in mW		
CUT	**CA**	**LFSR**
C_3540	1.107	1.729
C_432	1.537	1.625

LHCA for the reason that the switching activity due to the shifts is more and its structure includes as many Flip-flops as the number of bits are expected to be (Ex: 3 bits- 3 Flip-Flops, 8bits-8 Flip-Flops). But it strongly carries the advantage of maximum length sequence LFSR, which can be constructed using the primitive polynomials and being very difficult in the case of CA. The CA remains more random in nature, [6] when compared to the LFSR TPG for the reason that it doesn't involve correlation between the patterns generated, as it is to be with LFSR. Further a clear visual interpretation in reference to the Power (Total Power) and Area (Number of Cells) parameters is presented through Graphical Analysis in Figures 7, 8.

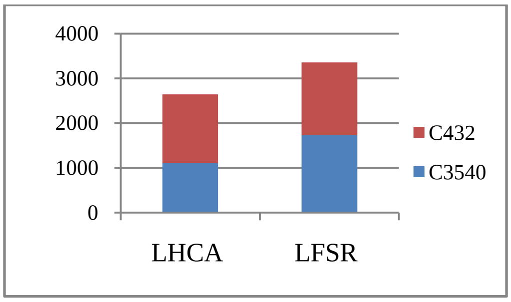


Figure 7. *Graphical Analysis of 'Total Power' parameter for all the two TPGs*

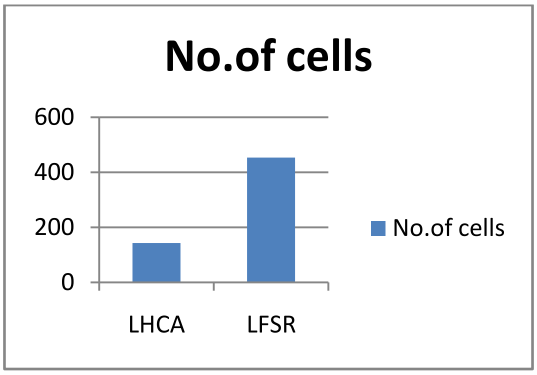


Figure 8. *Graphical Analysis of 'Number of Cells' parameter for all the two TPGs*

6. Conclusion

With increase in the complexity of the circuits and number of transistors on the chip, the need for a good TPG in order to implement the BIST capability and gain good extent of fault coverage through increased randomness in the patterns generated has been quite evident till now. Along with the need, also plays a major role are the parameters like Power, Area depending on which the TPGs have been analyzed. We have analyzed LHCA which is better when compared to LFSR, and to detect faults in DUT. Further study is in progress for detecting faults using Mobile Cellular Automata.

References

[1]. A New Kind of Science by Wolfram. Stephen, Wolfram Media Inc., 2002.

[2]. M. Abramovici, M. A. Breuer, A. D. Friedman, *Digital Systems Testing and Testable Design, IEEE Press, New York, 1990.*

[3]. C. Stroud. *A Designer's Guide to Built-In Self- Test.* Kluwer Academic Publishers. Boston M.A., 2002.

[4]. M.L. Bushnell. V.D. Agrawal. *Essentials of Electronics Testing for Digital, Memory &Mixed Signal VLSI Circuits,* Kluwer Academic Publishers, Boston M.A., 2000.

[5]. *What's an LFSR*, Texas Instruments-Literature No: SCTA036A/Support document.

[6]. Comparison of LFSR and CA for BIST by SachinDhingra, Dept of Electrical and Computer Engineering, Auburn University.

[7]. K.Salman, "Analysis of elementary Cellular Automata Boundary Conditions," International journal of Computer Science & Information Technology(IJCSIT) vol 5, No.4, August 2013.

[8]. P.R. Bardell. XV.H. McAnney. J. Savir. *Built-in Self-test for VLSI: Pseudorandom Techniques,* John Wiley and Sons. New York 1987.

[9]. P.D. Hortensius. R.D. McLeod. XV. Pries. D.M. Miller. H.C. Card. "Cellular automata-based pseudorandom number generators or built-in self-test" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 8 &pp 842-859. 1989.

[10]. Wei-Cheiig, Lien, Kuen-Jong Lee, Tong-Yu Hsieh, Wee-Lung Ang"An Efficient On-Chip Test Generation Scheme Based on Programmable and Multiple Twisted-Ring Counters "IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems. Vol. 32, No. 8, August 2013.

[11]. MehrdadNourani, Mohammad Tehranipoor, and Nisar Ahmed, Low-Transition Test Pattern Generation for BIST-Based Applications IEEE Transactions On Computers, Vol. 57, No. 3, March 2008.