International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 2, Issue 3, pp: (44-50), Month: July - September 2014, Available at: www.researchpublish.com

"CELLULAR AUTOMATA" FOR BUILT IN TEST PATTERN GENERATION AND TEST RESPONSE ANALYZER

[1]Harshala Patil, [2]Shyam Sunder Padhy

Abstract: **Increasing growth of sub-micron technology in the semiconductor industry has resulted in the difficulty of VLSI testing for designers who always look for simple, regular and cascadable logic structure to realize a complex function. A system composing of cells interconnected in a regular manner whose behavior advances in time in discrete steps is called Cellular Automata (CA). The main goal of this paper is to analyze CA for built in test pattern generation and analyze the test response, hence minimizing the problems related to VLSI testing. Further, Linear Feedback Shift Register (LFSR) and CA are compared. Also, efficient test patterns using CA are explained, along with a plan to improve energy conservation in CA and to generate a power test in order to check the fault coverage.**

Keywords: **Built in Self-Test, Cellular Automata, Linear Feedback Shift Register, Signature Srutinization, Test Pattern Generator and Analyzer.**

I. INTRODUCTION

In today's world there are many problems faced under VLSI circuit testing as traditional test techniques have become extremely expensive and no longer provide sufficient high fault coverage. Today's test requirements expect much higher flaw coverage, smaller test arrangement, shorter test cycles, at-speed test and better performance-cost ratio. Hence to detect un-modelled faults and provide remote diagnosis, Built-in Self-Test (BIST) is needed.

Implementation of BIST is done by Linear Feedback Shift Register (LFSR) and Cellular Automata (CA) where LFSR is a shift register with feedback path linearly related to the nodes using XOR gates and CA is a collection of nodes logically related to their neighbors using XOR gates. Test pattern generation based upon CA possess some advantages over traditional LFSR approaches [7].

A system composing of cells interconnected in a regular manner whose behavior advances in time in discrete steps is called a CA where connections between the neighboring cells are expressed as rules. These rules determine the next state based on the state of the cell's neighbors.

Cellular Automata has attracted considerable interest because each cell can only connect to its local neighbors. Therefore, unlike LFSR, where expansion in size requires major changes to feedback loop connections, CA are easily scalable. It is enough to simply connect more cells at the end of an existing CA. Also, the CA has a regular, modular and cascadable structure with local neighborhood interconnections that ideally suits VLSI technology [8].

II. BUILT IN TEST PATTERN GENERATION

I. BIST Architecture

Basic BIST architecture includes functions which are necessary to execute the self-testing feature so that testing is accomplished without the aid of external hardware. A typical BIST architecture and its working is illustrated in Fig. 1.

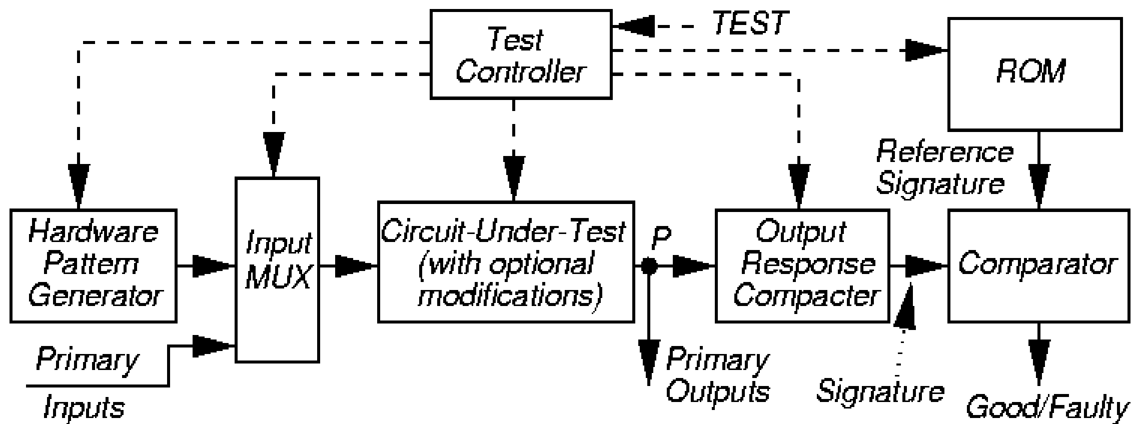

Fig. 1: BIST Architecture

It has two major components named Hardware or Test Pattern Generator (TPG) and Output Response Compacter or Analyzer (ORA). The TPG produces a sequence of patterns for testing the Circuit Under Test (CUT) while the ORA compacts the output responses of the CUT into some type of pass/fail indication which decides good or faulty result. The other two functions needed for system-level use of BIST include the test controller and input MUX. Besides the normal input/output (I/O) pins, the incorporation of BIST may also require additional I/O pins for activating the BIST sequence and to give valid results. The input test patterns can be stored in a Read Only Memory (ROM). Expected responses are read from ORA ROM and are compared to the actual output response of the CUT for each test vector. Any mismatch detected by the comparator is latched to indicate a failure has occurred during the BIST sequencing [9].

II. Test Pattern Generation for BIST

Test Pattern Generators are constructed from LFSRs. Exhaustive testing, pseudorandom testing and pseudo exhaustive testing are the three types of test pattern generation in BIST. Exhaustive test-pattern generators are used in exhaustive testing; weighted and adaptive test generators are used in pseudorandom testing while counters: syndrome driver & constant weight, combined LFSR and shift register, combined LFSR and XOR gates, cyclic LFSR etc. are used under pseudo exhaustive testing [2].

III. BIST Implementation

Implementation of BIST is done by Linear Feedback Shift Register or by Cellular Automata.

III.I Linear Feedback Shift Register

Standard LFSR consists of 'n' D flip flops and a selected number of XOR gates. In modular LFSR, each XOR gate is placed between two adjacent D flip flops. The internal structure of the n-stage LFSR can be described by a characteristic polynomial of degree n, f(x). On the basis of feedback, LFSR is classified into two types; external and internal as shown in Fig. 2. In a characteristic polynomial, all zero states are invalid having a maximum sequence length of $2^n - 1$. Characteristic polynomials are of two types; primitive and non-primitive. Reciprocal of a primitive polynomial is also primitive; $P^*(x) = x^n P(1/x)$. All LFSRs have a compact design with less than one gate per node. They follow a parallel pattern generation. Internal and external XOR are the two techniques of polynomial division used in LFSR.

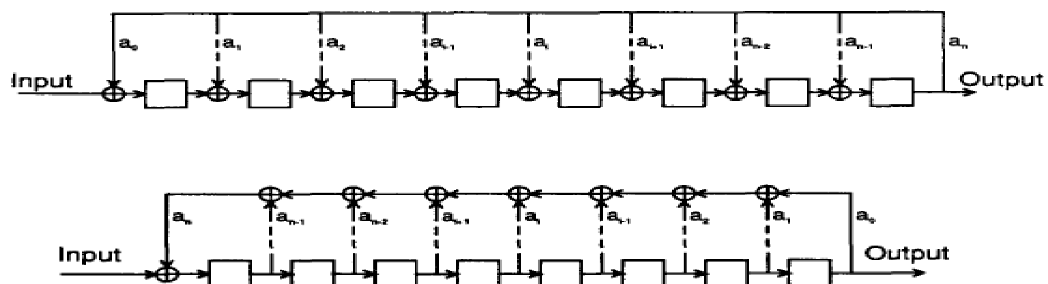

Fig. 2: Internal LFSR (top) and External LFSR (bottom)

By applying 2^n exhaustive patterns to an n input combinational CUT, exhaustive testing is done. Binary counter and complete LFSR contribute to the exhaustive pattern generator. Exhaustive testing guarantees all detectable, combinational faults to be detected.

Pseudo random testing gives pseudo random pattern generator. It reduces the test length but sacrifices the fault coverage. In this type of testing it is difficult to determine the required test length and fault coverage. Maximum length LFSR, weighted LFSR and Cellular Automata all follow the pseudo random testing. Pseudo exhaustive testing reduces time test while retains many advantages of exhaustive testing. It also guarantees 100% single-stuck fault coverage.

III.II Cellular Automata

A CA is an array of cells where each site is in any one of a few permissible states. At each clock cycle, the evolution of a site value depends on combinational logic, which is a function of the current state of k of its neighbors for a k-neighborhood automaton. A CA configured with rule 90 and rule 150 specifies evolutions from a neighborhood configuration to the next state as follows,

111	110	101	100	011	010	001	000	
0	1	0	1	1	0	1	0	decimal 90
1	0	0	1	0	1	1	0	decimal 150

For rule 90 the output depends on the left and right neighbor, while for rule 150, the output depends on the right and left neighbor and itself. A CA characterized by XOR and/or XNOR dependence is called an additive CA [6]. There are 16 such additive directive rules. If in a CA, the same rule is applied to all the cells, then the CA is called a uniform CA.

Example: A four-cell hybrid CA with null boundary condition with the rule vector<90 150 90 150> (i.e. <0 1 0 1>) applied from left to right may be characterized from a matrix T where 'a' is periodic boundary and 'b' is null boundary [1].

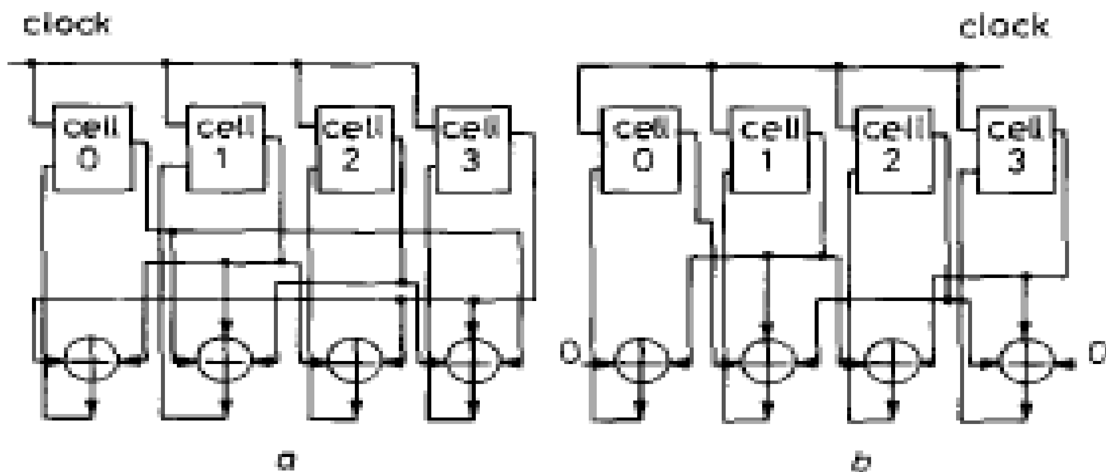

Fig. 3: 90/150 hybrid CA with rule vector <0 1 0 1>

CA provides more random test patterns and provides high fault coverage in a random pattern resistant circuit. It also gives an implementation advantage. If the test vectors generated by the CA are pseudo-random, the test set will contain appropriate vectors to active enough faults in CUT. A number of experiments have been performed to assess fault coverage. The result of the experiments is that stuck at fault coverage offered by the linear CA is satisfying, similar to the LFSR. Therefore, the test vectors generated by the CA are pseudo-random. Linear Hybrid Cellular Automata and Linear Cellular Automata Register are one-dimensional Linear CA. With null boundary condition with no feedback, the process is faster and with cyclic boundary condition with feedback the process is slower. CA uses highly random vectors.

III.III Comparison of LFSR and CA

LFSR and CA comparison tells us that the area overhead is least (less than one gate/node) in LFSR while higher than LFSR for CA (one gate/node). Maximum length sequence is easy to implement in LFSR while harder in CA since the combination of rules is not well defined. Performance is low in the external feedback LFSR and high in internal feedback LFSR. On the other hand, there are no gates involved in feedback which increases the CA performance. Shifting of data leads to low parallel pattern randomness in LFSR while logical relation with neighbors lead to high parallel pattern randomness in CA.

Although LFSRs are more popular because of their compact and simple design, CAs provide patterns with higher randomness. CAs performs better in detection of faults such as stuck-open or delay faults, which need two-pattern testing. Hence, CAs provides a good alternative for LFSRs when high fault coverage is needed.

III. TEST RESPONSE ANALYZER

I. Signature Srutinization

Srutinization of a signature or simply Signature Analysis is a method of circuit response compaction during BIST which reduces the huge number of bits in the original responses into a very small size that represents a statistical circuit property. However some information in the original response is lost during the process of compacting. As a result, there is a small probability that a signature of a faulty circuit may match that of the fault free circuit after response compaction. This problem is known as aliasing.

In signature analysis, we have a Signature Analysis Register (SAR) and Multiple Input Signature Register (MISR) as shown in Fig. 4. It is the most popular compaction technique used today based on cyclic redundancy checking. Serial signature analysis and parallel signature analysis are two signature analysis schemes. On defining a L bit output sequence M where M(x) = m0+m1x+m2x+…Ml-1x^L-1 signature is the polynomial remainder r(x) if M(x)=q(x)f(x)+r(x) where polynomial of the modular is f(x). In parallel signature analysis, MISR are used. An n-input MISR can be remodeled as a Single Input Signature Register (SISR) with effective input sequence M(x) and effective error polynomial E(x).

M(x) = M0(x) +xM1(x) +…x^n-2.Mn-2(x) +x^n-1.Mn-1(x)

E(x) = E0(x) +xE1(x) +…x^n-2.En-2(x) +x^n-1.En-1(x)

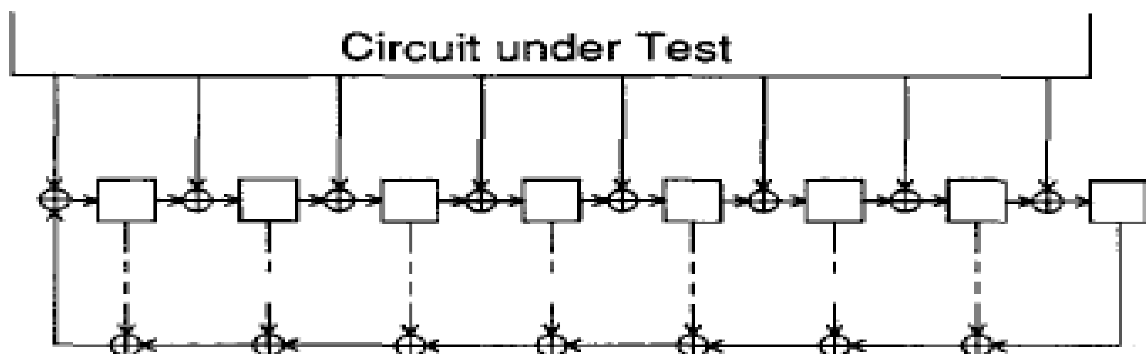

Fig.4: Multiple Input Signature Analysis Register

Error detecting and correcting circuits make extensive use of LFSRs. They are understood in algebraic coding as syndrome detection and in digital testing as signature analysis. Here we focus on the use of CA for signature analysis in BIST and how it is different from LFSR based SA.

II. LFSR Signature Analysis

The LFSR as a data compacter has already been researched extensively in many years. It is now a popular choice as a data compacter because of its ease of implementation and its degree of effectiveness. It can be implemented using the XOR gates and a set of Flip Flops. The operation of LFSR is related to the mathematics of polynomials. Every LFSR has a characteristic polynomial that describes its behavior. When the LFSR acts as a signature analyzer, the input data stream

polynomial is divided by LFSR characteristic polynomial. At the end of clock cycles, the last state of the LFSR or the remainder polynomial forms the signature of the circuit under test.

III. CA Signature Analysis

Usually, BIST methodologies employ LFSRs for the test vector generation. Often a string of pseudorandom test vector is required and in these situations a CA may prove to be preferable due to the significantly reduced cross- correlation of the bit streams [3]. Hence, one of the main motivations for considering signature analysis using CA is that, in cases where a CA is preferable for the test vector generation, it may be highly desirable to use the same CA for signature analysis as well. In this section, we proceed to describe and propose some measure on the effectiveness of signature analysis using CA.

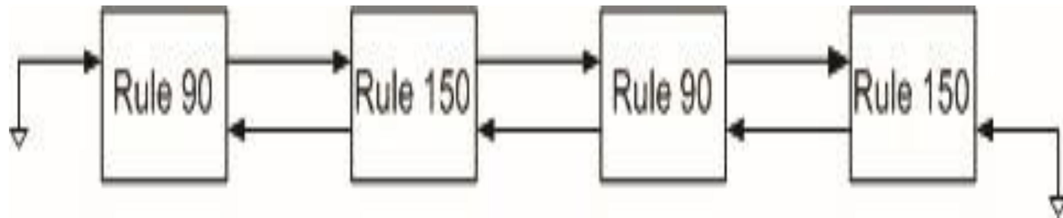

Fig. 5: Four-Stage Hybrid CA.

Fig. 5. Shows a four-stage hybrid CA where although linear hybrid 90/150 CA sequencing and the signature are different from that of LFSR, they have the same cycle structure. Therefore, those linear hybrid 90/150 CA have their aliasing probabilities of $2^{(-n)}$, which are same as the corresponding LFSR, when they act as single input response compactor.

IV. ALGORITHM TO TEST CA POWER CONSUMPTION

The Test Pattern Generator (TPG) in deterministic BIST often suffers from some problems such as extra test power consumption, area overhead and idle test cycles. Here, an efficient algorithm is discussed to synthesize a built-in TPG from low power deterministic test patterns without inserting any redundancy test vectors. The anatomy of TPG is based on the non-uniform CA and is used to test combinational circuits. The algorithm is depended on the nearest neighborhood prototype, which can find an optimal non-uniform CA topology to generate given low power reproduction of test. Simulation results using referenced combinational circuits show that the generator is efficient to generate the deterministic test patterns in terms of power utilization, area overhead and test time [5].
The main process of the algorithm is as follows,

Step1: Set the initial solutions such as the precompiling low power test pattern set to the nearest neighborhood matrix.
Step2: Check the evolution property based on Von Neumann neighborhood. If violation exists, execute step 3, else step 2 and check next cell.
Step3: Adopt the nearest 3-neighborhood function to obtain the optimal solution. If violation exists, go to the step 4, else back to step 2 for next cell.
Step4: Use the 5-neighborhood model to validate the evolution. If evolution satisfies then come back to step 2.
Step5: Obtain the interconnection and appropriate information.
Step6: Compute rule function and end program.

The algorithm can balance the trade-off between wiring topology and CA structure uniformity under the constraint of test power consumption.

V. ENERGY CONSERVATION IN CELLULAR AUTOMATA

Wireless sensor nodes are severely constrained by energy consumption as they are battery powered. Maximum types of control techniques that are used for energy conservation assume that the sensor nodes are uniformly established generally in form of grids. In real synopsis, a grid based deployment presents its own set of problems and limitations. Here we discuss a block CA based energy conservation technique.

Depending upon the neighborhood radius where the radius mentioned is basically the number of cell, a 3x3 Moore type neighborhood or a neighborhood of 8 members is considered as a standard [4]. A polygon can have varying number of neighbors in its neighborhood ranging from 3 to 14, with an average of 6. It was decided to keep the standard to 8 neighbors, so that the changeover rule for the focal cluster would be resolute. While searching for adjacent neighbors there can be a few scenarios where N = number of adjacent neighbors while N_{max} = maximum permissible limit of neighbors:

1. N=8, Moore neighborhood, the BCA based neighborhood rule, can be applied effectively.
2. $N>8<N_{max}$, the neighbors exceeds 3x3 Moore neighborhood, a little more than the standard neighborhood will not harm the rule's logic.
3. $N> N_{max}$, the new neighbors should be selected in decreasing order of their mean cluster energy, i.e. the cluster with maximum mean energy gets selected first and then the next best cluster and similarly the next.
4. N<8, the number of neighbor is less than the standard hence the neighborhood radius is increased by one unit and then checked again if the neighborhood has improved (see Fig. 6.).

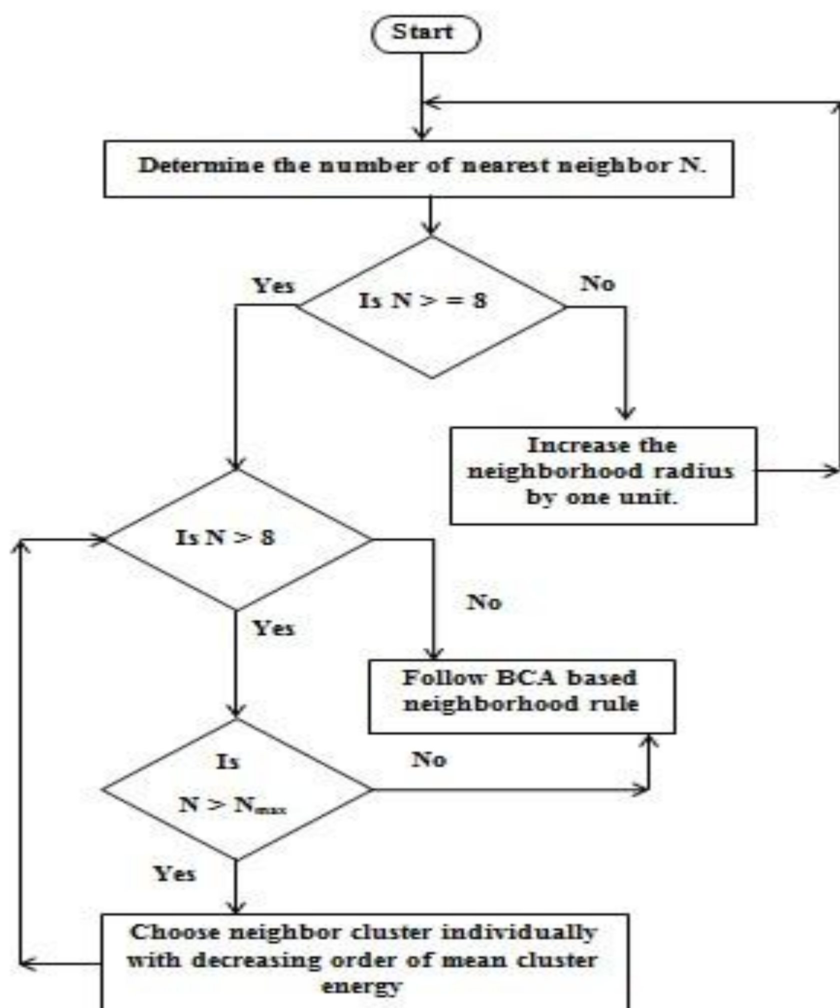

Fig. 6: Flowchart Showing The Selection Of Neighbors From Non-Uniform Neighborhood.

Any study or experimental data on the maximum allowable limit of neighbors for an irregular CA (for better stability) could not be found. So the value of N_{max} is undetermined but can be possibly assigned a value of around 12 to 14.

VI. CONCLUSION

Since each cell can only connect to its local neighbors, CA has attracted considerable interest when compared to LFSR. Also, CA are easily scalable as one can easily connect more cells at the end of an existing CA. Thus we can say that CA

minimizes problems like high fault coverage and low test time which are some of the VLSI testing concerns. Since powered by battery, wireless sensor nodes are severely constrained by energy consumption. Improvement over a block CA in wireless sensor network for energy conservation is discussed. Further, an algorithm to find an optimal CA structure that will generate deterministic low power test sets and cover high fault coverage check is explained.

REFERENCES

[1] S. Nandi, B. Vamsi, S. Chakraborty, P. Pal Chaudhuri, "Cellular automata as a BIST structure for testing CMOS circuits," IET Computers and Digital Techniques, IEE Proceedings, Volume: 141, Issue: 1, pp.41-47, Jan 1994.

[2] Lixin Gao, Yongliang Zhang, Jinhong Zhao, "BIST using Cellular Automata as Test Pattern Generator and Response Compaction," IEEE Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference, pp. 200-203, 21-23 April 2012.

[3] Peter D. Hortensius, Robert D. McLeod, Howard C. Card, "Cellular Automata-Based signature analysis for Built-In Self-Test," Computers, IEEE Transactions, Volume: 39, Issue: 10, pp. 1273-1283, Oct 1990.

[4] Chayan Banerjee, "Energy conservation in wireless sensor network using irregular cellular," IEEE Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference, pp. 223-227, 10-12 April 2013.

[5] Bei Cao, Liyi Xiao, Yongsheng Wang, "A low power deterministic test pattern generator for BIST based on cellular automata," IEEE Electronic Design, Test and Applications, 2008. Delta 2008. 4th IEEE International Symposium, pp. 266-269, 23-25 Jan 2008.

[6] Parimal Pal Chaudhuri, "Additive Cellular Automata: Theory and Applications, Volume 1."

[7] Stanley Leonard Hurst, "VLSI testing: Digital and Mixed Analogue/ Digital Techniques".

[8] George W. Zobrist, "VLSI Fault Modelling and Testing Techniques".

[9] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, "VLSI Test Principles and Architectures: Design for Testability".