BOUNDARY SCAN WITH CELLULAR-BASED BUILT-IN SELF-TEST

CLAY S. GLOSTER, JR.[1,3]

FRANC BRGLEZ[2,3]

[1]North Carolina A&T State University

[2]Bell Northern Research

[3] Microelectronics Center of North Carolina
P. O. Box 12889
Research Triangle Park, NC 27709
(919) 248-1943

Abstract

We discuss an approach to merging Boundary Scan with Built-In Self-Test. The proposed implementation of Boundary Scan represents a snap-shot of the Joint Test Advisory Group Recommendation 1.0, while the Built-In Self-Test implements the features of cellular automata. We examine test patterns generated from two distinct sources, one with registers using cellular automata and the other, based on the conventional LFSR configuration. We analyze and illustrate distinctive effects of these patterns on fault coverage of specific designs.

1 Introduction

As the complexity of designs increases, so does the problem of testing. Today's manufacturers of integrated circuits typically use in-circuit and functional board test systems to detect defects in their products. In-circuit tests are applied directly to the pins of each component, while functional tests use the board edge connector. These two techniques have been used separately or in sequence.

Recently, there has been an increasing trend in the use of surface mount package designs. Surface mount technology (SMT) allows reduced pin spacing as well as a higher density of packages on a printed circuit board. With this new technology, bed of nails fixtures for in-circuit tests have become less cost effective, thus introducing the need for advances in testing techniques.

Boundary Scan is a testing technique that allows the circuit to be tested via the board edge connector. It introduces a shift register that is logically, and often physically, adjacent to the I/O pins of every chip on the board. The shift register is used to shift, apply, or capture test data and can thereby test, not only individual chips, but also the board interconnect.

There has been an industry-wide effort to introduce and standardize boundary scan. The Joint Test Advisory Group(JTAG), with members from several companies, has presented several recommendations towards this standard[1] , [2]. The proposed standard defines and justifies various modes of boundary scan and offers guidelines for implementation. The proposal does not exclude a framework for Built-In Self-Test within Boundary Scan.

The paper begins with an overview of a Boundary Scan Template which includes Built-In Self-Test (BIST) mode within the context of JTAG Recommendations. We next examine the specific input register design and project area cost for various size pad frames. A major section of the paper analyzes test patterns generated from CA-based registers as well as LFSRs.

2 Overview

Several proposals and specific implementations of Boundary Scan that also integrate BIST have been published earlier [3], [4], and [5]. The purpose of this work is to examine JTAG recommendations 1.0[2] and design a boundary scan template that can be integrated with an existing scan-based design. A distinctive feature of the template is the Built-In Self-Test(BIST) mode where register designs are based on principles of cellular automata[6]. Figure 1 shows a block diagram of boundary scan and it's primary interfaces to the chip interior. It should be noted that during our work, JTAG recommendations were a moving target. However, we do not foresee any major difficulties in adapting the results of the current work to the latest JTAG recommendations [7].

The template consists of an input register, an output register, a mode register, an output multiplexer, and some additional control logic. Two additional control pins *BSE* and *BAS* along with two scan pins *BSI* and *BSO* are required. The registers in this template accommodate all of the test modes proposed by JTAG along with an added built-in self-test mode as described below.

The principal modes of boundary scan are, normal, scan and apply test modes. In the normal mode of operation, the circuit performs it's nominal function. The scan mode allows data to be shifted in or results to be shifted out. The apply test mode is divided into several sub-modes that perform other tests including BIST.

Once the apply test mode is entered, the mode register is used to determine the sub-mode of operation. There are 5 sub-modes associated with the apply-test mode. These

CH2610-4/88/0000/0138$01.00 © 1988 IEEE

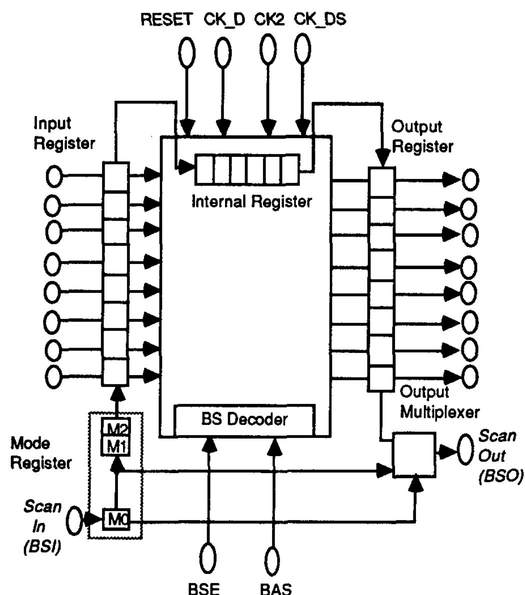

Figure 1: The Structural Template for Boundary Scan with BIST

are external test, internal test, sample test, bypass, and our added built-in self-test mode.

External test is used to test the interconnect of the printed circuit board. Data is applied to the printed circuit board from the output register. The input register latches the data that is flowing from another chip via the printed circuit board. This data can then be shifted out for verification.

Internal test provides the means to test the internal logic of the design. Data is applied from the input register to the circuit. The corresponding responses are latched in the output register. Once again, the results can be shifted out and verified.

Sample test allows the test engineer to take a snap-shot of the circuit at a particular instant of time. Data is latched in both the input and output registers.

Bypass mode makes use of the output multiplexer to bypass the chip's lengthy boundary scan path. Without this feature, testing a board with 100 chips, each containing 100 I/O pins, would take a considerable amount of time. Once placed in bypass mode, data on the chip travels from the *BSI* pin, through one latch, and directly to the *BSO* pin.

In the built-in self-test mode, the input register is reconfigured to a pseudo-random pattern generator, while the output register functions as a signature analyzer. Random patterns are shifted serially into the internal scan register and then are applied synchronously with the ones from the input register. The responses from these random patterns are compressed in the output register. A resulting signature can be checked to ensure proper circuit operation.

3 Input Register Design

This section presents the implementation of the input register. The output register is very similar to the input register except that it operates as a multiple input signature analyzer instead of a pattern generator. A more detailed description of all of the proposed hardware implementation is discussed in [8].

Figure 2 shows how the register is reconfigured during various test modes. The input register must appear transparent in the normal mode of operation, it must latch the data during external and sample mode, it must form a scan chain during scan mode, it must be able to apply data in the internal mode, and it must generate pseudo-random patterns in the BIST mode.

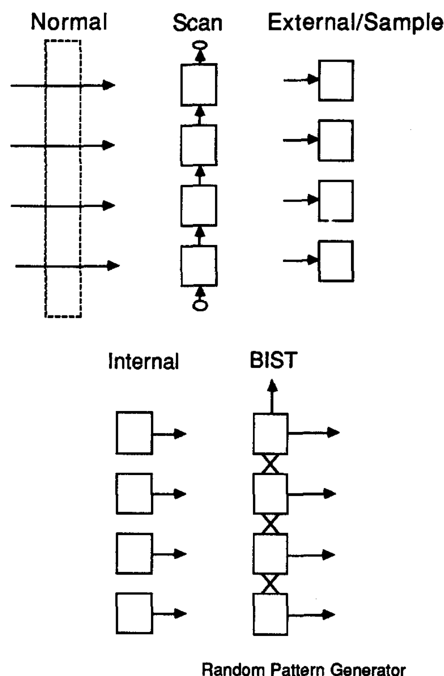

Figure 2: Input Register Modes (4-bit example)

The register was designed by combining the boundary scan input cells as recommended by JTAG while incorporating the principal of cellular automata [6]. While differing in details of implementation, several functions of this register are similar to the BILBO register[9]. Since approaches to generating pseudo-random patterns using cellular automata are relatively new [6], we provide basic details here.

A cellular automaton evolves in discrete steps with the next value of one site determined by it's previous value and that of a set of sites called neighbor sites. That is, the next value of the present cell is dependent on the previous value of the cell to it's left and right. The CA may be cyclically connected, or it may possess null boundary conditions. The

null boundary condition was used in this research. This configuration removes the need for a lengthy feedback loop between the first and the last cell.

It has been shown in [6] that by combining cellular automata rules 90 and 150 one can generate maximal length binary sequences from each site. That is, the combination of rule 90

$$a_i(t+1) = a_{i-1}(t) \oplus a_{i+1}(t)$$

and rule 150

$$a_i(t+1) = a_{i-1}(t) \oplus a_i(t) \oplus a_{i+1}(t),$$

where "i" is the index of cell "a", can yield a maximal length sequence of $2^s - 1$, s being the number of cells or the length of the CA. The construction rules that yield a CA register with maximal length sequence have been proposed in [6] and are reproduced in Table 1. We note that results based on CA registers in Table 1 can be readily compared with maximal length LFSR-based configurations [10], [11], [12].

Length	*Construction Rule**	*Cycle Length*
4	0101	15
5	11001	31
6	010101	63
7	1101010	127
8	11010101	255
9	110010101	511
10	0101010101	1,023
11	11010101010	2,047
12	010101010101	4,095
13	1100101010100	8,191
14	01111101111110	16,383
15	100100010100001	32,767
16	1101010101010101	65,535
17	01111101111110011	131,071
18	010101010101010101	262,143
19	0110100110110001001	524,867
20	11110011101101111111	1,048,575
21	011110011000001111011	2,097,151
22	0101010101010101010101	4,194,303
23	11010111001110100011010	8,388,607
24	111111010010110101010110	1,677,7213
25	1011110101010100111100100	33,554,431
26	01011010110100010111011000	67,108,863
27	000011111000001100100001101	134,217,727
28	0101010101010101010101010101	268,435,455

Construction Rules*

Configuration 1	Configuration 2
"0" represents a rule 90 cell	"0" represents a rule 150 cell
"1" represents a rule 150 cell	"1" represents a rule 90 cell

Table 1: CA Register Configurations

An example is shown in Figure 3. The input register is placed between the input pins and the circuit logic. A maximal length sequence for 4 inputs is obtained by alternating Rule 90 (odd) and Rule 150(even) cells.

The signals IE and SHF are the control signals that place the register in one of the configurations shown in Figure 2. The register is placed between the input pins and the circuit logic and is also serially connected to the internal scan register.

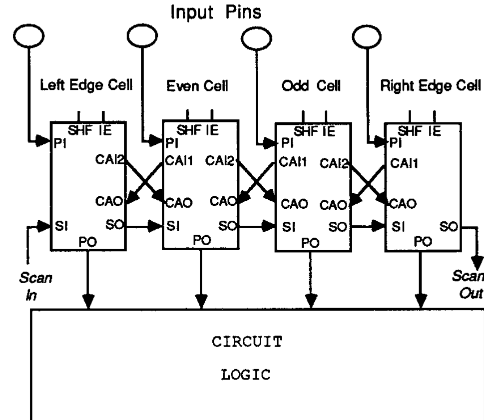

Figure 3: Input Register Design (4-bit example)

Figure 4 shows a sample input cell. It consists of 2 multiplexers, a scannable flip-flop, and an exclusive-or gate for pattern generation. This is a rule 90 cell as the next value of this cell is equal to the exclusive-or of the value of the cell to it's left with the value of the cell to it's right.

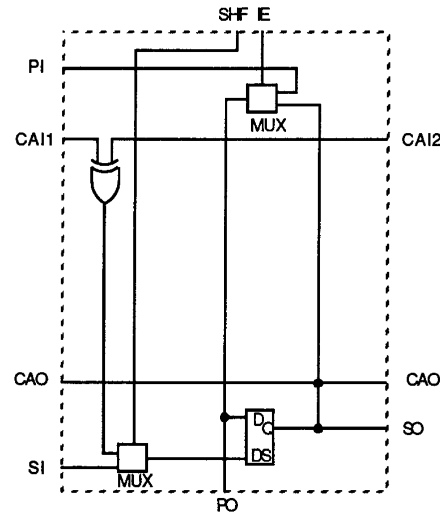

Figure 4: Realization of a Rule 90 Input BS Cell

4 Cost Assessment

This section presents a projection of the overhead that is required for inclusion of Boundary Scan with Built-In Self-Test within an existing design. Cost projections were made based on the size of our largest unidirectional cell.

Our initial implementation for functional verification of our largest cell contained 3 exclusive-or gates, 2 multiplexers, 1 demultiplexer, and a scannable flip flop. This implementation required a total of 13 logic gates or standard cells. The design consumed a total of 91 transistors and, with loose wiring, an area of 255 sq μm using a 3μm CMOS technology.

Optimization of the largest cell reduces the number of transistors needed to 68. Custom design of these cells will decrease the area, but it is anticipated that feedthroughs will be needed which may offset this decrease in size. Thus, an analysis of required chip area was performed using the conservative area estimate above.

Since there will be a boundary scan cell for each primary input and output, we suggest that the cells be placed adjacent to the pins of the design. This literally constitutes a "peripheral scan", as the cells would be placed on the periphery of the design. Figure 5 shows the projected location of all required hardware to achieve boundary scan and BIST.

The template consists of the boundary scan cells as well as some additional control logic. Note that, in some instances, some area available in the shaded region, nominally reserved for boundary scan cells, may accommodate the additional control logic.

If we are to include the boundary scan cells in the shaded region of Figure 5, we must ensure that the width of the cells is less than the distance between adjacent pins. We found that our largest cell could fit along side of a pin with additional area available for routing.

Several pad frames were analyzed to estimate the cost of testability in terms of chip area. These results are shown in Table 2. The maximum usable area before boundary scan is the area of the frame minus the area of the pads. We calculated our maximum usable area after boundary scan by placing our largest cell along side of each I/O pin. The table shows that for large frames, the decrease in usable area is relatively small.

It is anticipated that cells that could handle bidirectional data would be constructed from the union of an input and an output cell. Because of the constraint in pin spacing, these cells would become rectangular and would increase overhead. As all of these cost projections were done for a 3 μm technology, smaller technologies could be used which would provide space for these larger cells.

Another possibility to reduce the cost of the proposed scheme would be to merge the pads with the boundary scan cells such that each of the above pad frames would have a 0% decrease in maximum usable area. Automation of placement and routing of these cells would also decrease design time and possibly, the related costs.

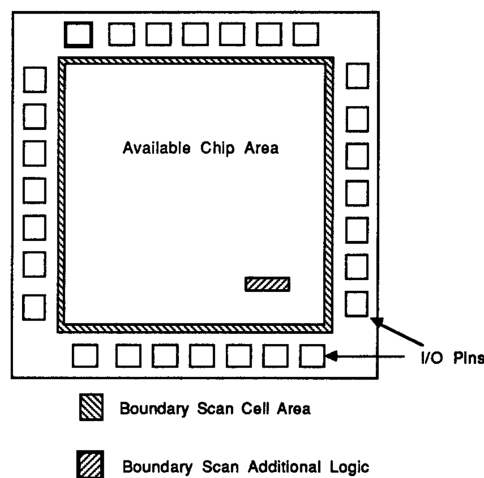

Figure 5: An Effective Location for Boundary Scan with BIST

Pins	*Frame Size*	*Usable Area Before Boundary Scan*	*Usable Area After Boundary Scan*	Calculated Overhead
28	S	11.8 mm^2	10.1 mm^2	16.96%
40	S	11.8 mm^2	10.1 mm^2	16.96%
40	M	25.7 mm^2	23.2 mm^2	11.15%
40	L	40.2 mm^2	37.1 mm^2	8.55%
64	M	25.7 mm^2	23.2 mm^2	11.15%
64	L	40.2 mm^2	37.1 mm^2	8.55%
64	XL	64.3 mm^2	60.2 mm^2	6.70%
84	L	40.2 mm^2	37.1 mm^2	8.55%
84	XL	64.3 mm^2	60.2 mm^2	6.70%

Table 2: Projected Overhead(including BIST) for Various Size Pad Frames

5 Evaluating Options with BIST

During evaluation of test patterns for designs that incorporate boundary scan and BIST, we were prompted to examine them further. This section is divided into two major parts. The first part introduces notions, illustrated with simple examples, that we found useful for explaining results with a design example in the second part.

5.1 The BIST Model

The key parameters in our BIST model are shown in Figure 6. These include the length "s" of the source register, number of PIs "n" and the length of the interior scannable register (latches) "m". The source register is an extension of the boundary scan register that already requires "n" cells for every PI. Clearly, s $\geq$ n. The additional cells in the source register may be required to meet the random pattern testability specification "N_{100}" of the circuit under test. This will be illustrated for a design example later on.

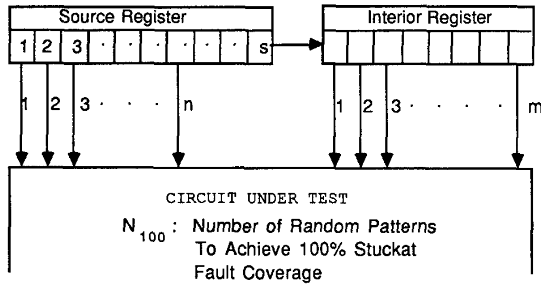

Figure 6: Characteristic Parameter Set (n,m,s) in the BIST Proposal

5.2 The Test Pattern Generation Process

This process begins by clocking the initialized source register "m times"; to load the interior register and then applying an (n+m)-wide "trial pattern" in a single clock cycle. Every trial pattern produces a potential "test pattern" the first time it is applied to the circuit under test. We enumerate this process in terms of the following variables:

$$\begin{aligned} x &= number\ of\ source\ patterns \\ y &= number\ of\ trial\ patterns \\ z &= number\ of\ test\ patterns \\ &\quad (number\ of\ trial\ patterns\ that\ are\ unique) \\ \eta &= z/y \\ &\quad (trial\ pattern\ efficiency) \end{aligned} \tag{1}$$

By also defining

$$\begin{aligned} T_z &= period\ of\ the\ source\ patterns \\ B_z &= maximum\ number\ of\ unique\ trial\ patterns \\ &\quad (test\ pattern\ bandwidth) \\ B_y &= number\ of\ trial\ patterns\ when\ z = B_z \\ &\quad (trial\ pattern\ bandwidth) \end{aligned} \tag{2}$$

we have simple measures to evaluate test patterns produced either by a LFSR-based or a CA-based source register. We also note that the following inequalities hold true:

$$\begin{aligned} T_z &\leq 2^s - 1 \\ B_y &\leq T_z \\ B_z &\leq B_y \end{aligned} \tag{3}$$

Clearly, for $s \geq 20$, direct evaluations of B_y and B_z will rapidly become unproductive and we must rely on other means to estimate values of these parameters. In order to meet random pattern testability specification N_{100} as shown in Figure 6, we must maintain

$$N_{100} < B_z \tag{4}$$

Figures 7 and 8 illustrate that even for small configurations, parameters such as (4, 2, 6) and (5, 1, 6), trial pattern bandwidth B_y, test pattern bandwidth B_z and trial pattern efficiency "η" may vary over a wide range of values; depending not only on specific values of (m, n, s) but also on the choice of the CA-based or LFSR-based register. Results obtained for a wider range of parameter values are summarized in Table 3. In this table, we also observe an additional phenomenon. Namely, if both 2^s-1 and m are divisible by a common factor "r", then the upper bound on the trial pattern bandwidth B_y will be reduced to $(2^s - 1)/r$. This phenomenon can further reduce the number of patterns that are available to test the circuit. The case in point in Table 3 takes place for values of m=3, where trial pattern bandwith has been reduced from 63 to 21. The polynomial we use for the LFSR part in Table 3 is $1 + x + x^6$. Finally, note that overall, pattern efficiency in CA-based configurations is equal to or consistently exceeds pattern efficiencies of LFSR-based configurations.

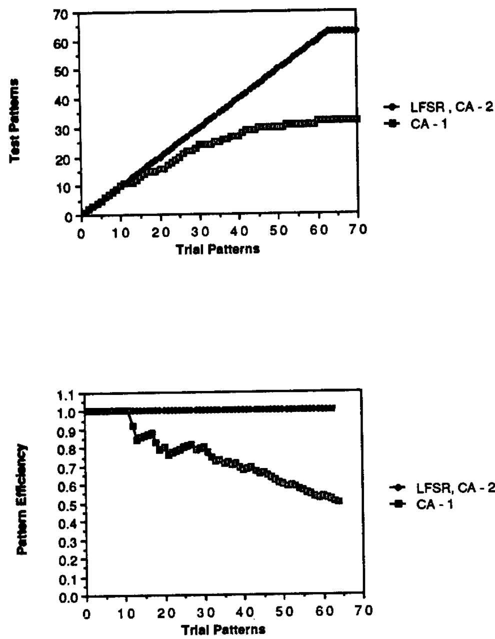

Figure 7: CA vs LFSR Characterization (n=4,m=2,s=6)

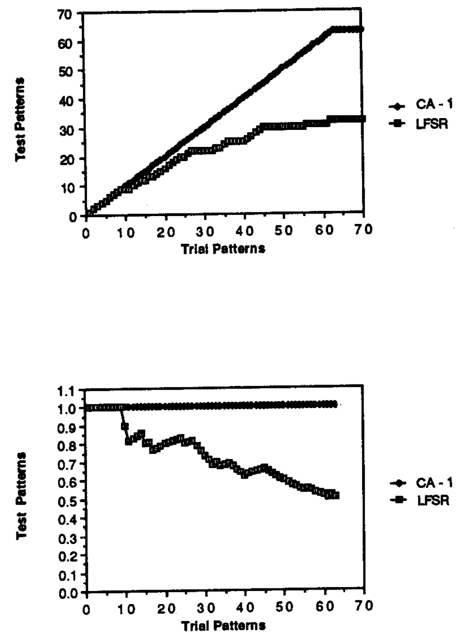

Figure 8: CA vs LFSR Characterization (n=5,m=1,s=6)

Number of Inputs (n)	*Interior Register Length (m)*	*Source Register Length (s)*	CA-Based Register			LFSR-Based Register		
			Trial Pattern Bandwidth	*Test Pattern Bandwidth*	*Pattern Efficiency*	*Trial Pattern Bandwidth*	*Test Pattern Bandwidth*	*Pattern Efficiency*
6	1	6	63	63	1.0	63	63	1.0
	2		63	63	1.0	63	63	1.0
	3		21	21	1.0	21	21	1.0
	4		63	63	1.0	63	63	1.0
5	1	6	63	63	1.0	63	63	1.0
	2		63	63	1.0	63	63	1.0
	3		21	21	1.0	21	21	1.0
	4		63	63	1.0	63	63	1.0
4	1	6	58	32	0.551	49	31	0.633
	2		63	63	1.0	62	32	0.516
	3		21	21	1.0	21	18	0.857
	4		63	63	1.0	52	32	0.615

Table 3: CA versus LFSR-based Test Pattern Generation, Small Examples

5.3 A Controller Design Example

We applied boundary scan and BIST techniques to an existing scan-based chip design [13]; it has a relatively small number of interior scannable latches (m=6) with respect to the number of inputs (n=16). By way of fault simulation with computer-generated random patterns [14], we found that the circuit requires 131,040 random patterns to achieve 100% single stuck-at fault coverage, i.e., the circuit random pattern testability specification is N_{100} = 131,040. According to (3) and (4), we must choose a source register with a value of at least s=18 in order to meet the random pattern testability specification. Consequently, we added two additional cells to the boundary scan register and began generating and fault simulating test patterns in two configurations: one with CA-based register as the source of random patterns and the other with LFSR. We quickly found that the CA-based configuration generated a 100% test within 41,888 patterns while fault coverage of LFSR-based test flattened after about 40,000 trials at 98.28%. It was this result that prompted us to examine the issue of test pattern generation in the direction discussed in the preceding section. The explanation we offer for poor performance of the LFSR-based test patterns in this case follows the notation and terminology of the simple example cases from Table 3. The results are summarized in Table 4. We now understand that since the period of the source $(2^{18}-1)$ and the size of interior register (m = 6) both have a common divisor of 3, the bound on the trial pattern bandwidth has been reduced to 87,381 for both CA-based and LFSR-based configuration. Subsequent analysis reveals that test pattern bandwith with the CA-based register also reaches 87,381 patterns while it does not exceed 56,603 patterns with the LFSR-based register. There are intrinsically less unique patterns to test the circuit in this LFSR configuration, hence there is less opportunity to reach full fault coverage in a circuit that is random pattern resistant. Only by increasing the size of LFSR to 19 do we find this circuit to be 100% testable.

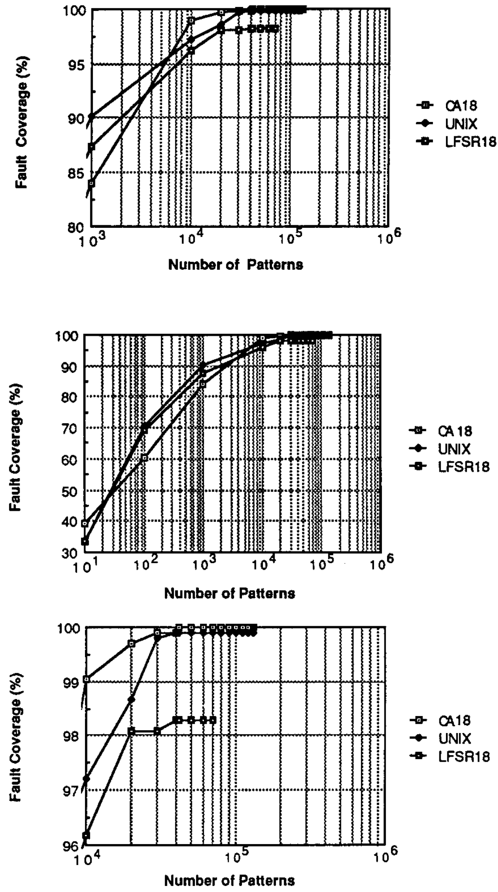

Figure 9: Fault Coverage Curves for CC1 Controller

Number of Inputs (n)	*Interior Register Length (m)*	*Source Register Length (s)*	CA-Based Register			LFSR-Based Register		
			Trial Pattern Bandwidth	*Test Pattern Bandwidth*	*Pattern Efficiency*	*Trial Pattern Bandwidth*	*Test Pattern Bandwidth*	*Pattern Efficiency*
16	6	18	87381	87381	1.0	87381	56603	0.602

Table 4: CA versus LFSR-based Test Pattern Generation, Main Controller Example

6 Summary and Current Work

We have shown that boundary scan can be realized with an acceptable overhead while offering a variety of test modes that are required to test high performance boards. In addition, we introduce a built-in self-test mode based on the principles of cellular automata.

We have shown that a CA-based generator may be more appropriate for BIST than the conventional LFSR-based generators. In addition to better randomness properties, the CA-based pseudo-random test pattern generator also has implementation advantages in that it requires only adjacent neighbor communication and is therefore cascadable, i.e., the physical length of the generator can be increased or decreased by simply adding or removing cells. Therefore, the major redesign required in the case of the LFSR is avoided. The modularity of a CA-based test pattern generator or signature analyzer makes them also very attractive for embedding into a modular CAD/CAT tool environment.

Our current work is directed towards optimizing and automating the layout of boundary scan cells with BIST (CA-based or LFSR-based), updating the register and controller design template to meet the JTAG 2.0 specifications, and further characterizing the random test pattern generation process as well as the signature analysis. We are expanding this work also in terms of the transition fault model; early results in evaluating transition fault coverage as defined in [15] point towards significantly higher transition fault coverage results when using CA-based random pattern generators.

Acknowledgements

We gratefully acknowledge Rod Tulloss from AT&T Engineering Research Center for keeping us updated with versions of JTAG Recommendations, Peter Hortensius from University of Manitoba for sharing with us early results from his PhD dissertation on random number generation with cellular automata, BNR for supporting a summer student position at MCNC in 1987, and NTI for supporting an Industrial Affiliate position at MCNC in 1988.

References

[1] F. Beenker and C. Maunder. Boundary-Scan, A Framework for Structured Design-For-Test. In *IEEE International Test Conference*, pages 724 – 729, September 1987.

[2] C. M. Maunder F. P. M. Beenker and C. Vivier. A Standard Boundary Scan Architecture Version 1.0. June 1987.

[3] J. J. LeBlanc. LOCST: A Built-In Self-Test Technique. *IEEE Design and Test*, November 1984.

[4] P. Bardell and W. McAnney. Self-Testing of Multichip Logic Modules. In *Digest of Papers 1982 International Test Conference, IEEE*, pages 200 – 204, November 1982.

[5] D. Komonytsky. LSI Self-Test Using Level Sensitive Scan Design and Signature Analysis. In *Digest of Papers 1982 International Test Conference*, pages 414 – 424, November 1982.

[6] Peter D. Hortensius. *Parallel Computation of Non-Deterministic Algorithms in VLSI.* PhD thesis, University of Manitoba, 1987.

[7] *JTAG Boundary-Scan Architecture Standard Proposal Version 2.0.* March 1987.

[8] C. Gloster and F. Brglez. *Integration of Boundary Scan with Cellular-Based Built-In Self-Test for Scan-Based Architectures, Version 1.0.* Technical Report TR87-18, Microelectronics Center of North Carolina, Research Triangle Park, NC, August 1987.

[9] B.Koenenmann, J. Mucha, and G. Zwiehoff. Built-In Logic Block Observer. In *Digest of Papers 1979 Test Conference, IEEE*, pages 37 – 41, October 1979.

[10] Don Lancaster. *TTL Cookbook.* Howard W. Sams & Co., 1974.

[11] Laung-Terng Wang and Edward McCluskey. Hybrid Designs Generating Maximum-Length Sequences. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(1):91 – 99, January 1988.

[12] William H. McAnney Paul H. Bardell and Jacob Savir. *Built-In Test for VLSI: Pseudorandom Techniques.* John Wiley and Sons, Inc., New York, 1987.

[13] G. Kedem and J. L. Ellis. The Ray Casting Machine. In *IEEE Proceedings ICCD '84*, pages 533 – 538, October 1984.

[14] J. Calhoun, D. Bryan, and F. Brglez. *Automatic Test Pattern Generation (ATPG) for Scan-Based Digital Logic: Version 1.0.* Technical Report TR87-17, Microelectronics Center of North Carolina, Research Triangle Park, NC, August 1987.

[15] F. Brglez and M. H. Schulz. Accelerated Transistion Fault Simulation. In *Proc. 24th Design Automation Conference*, pages 237 – 250, June 1987.