University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers | Computer Science and Engineering, Department of

2003

Low-Energy BIST Design for Scan-based Logic Circuits

Bhargab B. Bhattacharya
Indian Statistical Institute, Kolkata, India, bhargab@isical.ac.in

Sharad C. Seth
University of Nebraska - Lincoln, seth@cse.unl.edu

Sheng Zhang
University of Nebraska-Lincoln, szhang@cse.unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Sciences Commons

Bhattacharya, Bhargab B.; Seth, Sharad C.; and Zhang, Sheng, "Low-Energy BIST Design for Scan-based Logic Circuits" (2003). *CSE Conference and Workshop Papers*. 10.
http://digitalcommons.unl.edu/cseconfwork/10

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Low-Energy BIST Design for Scan-based Logic Circuits

Bhargab B. Bhattacharya[1], *Sharad C. Seth, and Sheng Zhang*

Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115, USA
(bhargab, seth, szhang)@cse.unl.edu

Abstract

In a random testing environment, a significant amount of energy is wasted in the LFSR and in the CUT by useless patterns that do not contribute to fault dropping. Another major source of energy drainage is the loss due to random switching activity in the CUT and in the scan path between applications of two successive vectors. In this work, a new built-in self-test (BIST) scheme for scan-based circuits is proposed for reducing such energy consumption. A mapping logic is designed which modifies the state transitions of the LFSR such that only the useful vectors are generated according to a desired sequence. Further, it reduces test application time without affecting fault coverage. Experimental results on ISCAS-89 benchmark circuits reveal a significant amount of energy savings in the LFSR during random testing.

I. Introduction

With the emergence of mobile computing and communication devices, design of low-energy VLSI systems has become a major concern in circuit synthesis. A significant component of the energy consumed in CMOS circuits is caused by the total amount of *switching activity* (SA) at various circuit nodes during operation. The energy dissipated at a circuit node is proportional to the total number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions the logic signals undergo at that node multiplied by its capacitance (which depends on its fanout, and its transistor implementation). Energy consumption in an IC may be significantly higher during testing due to increased SA than that needed during normal (system) mode, which can cause excessive heating and degrade circuit reliability. The average-power optimization help extend the battery life in mobile applications. The maximum-power, sustained or instantaneous, may cause excessive heating or undesirable logic swing. Conventional BIST schemes with random patterns may need an excessive amount of energy because of the test length and randomness of the consecutive test vectors. Further, a significant amount of energy may be wasted during just the scan operations.

[1]On leave from the *Indian Statistical Institute,* Calcutta – 700 108.

II. Related work

With the emergence of deep submicron technology, minimizing power/energy consumption during testing has become an important goal of BIST design. The existing techniques include toggle suppression and blocking useless patterns at the inputs to the CUT [1], designing low-power TPG [2], and low-transition random pattern generator [13], distributed BIST [17] and test scheduling [3], optimal vector selection, designing new scan path architecture [11], use of Golomb coding for low-power scan testing with compressed test data [14], and BIST for data paths [4]. For deterministic testing, energy reduction can be achieved by reordering scan chains and test vectors [6], or by test compaction [5].

III. New BIST design

We assume a *test-per-scan* BIST scheme as in the STUMPS architecture (Fig. 1). A *modulo-m* bit-counter keeps track of the number of scan shifts, where *m* is the length of the longest scan path. Since the number of useful patterns is known to be a very small fraction of all generated patterns, a significant amount of energy is still wasted in the LFSR while cycling through these useless patterns even though they are blocked at the inputs to the CUT [1]. Further, test-vector reordering in a pseudorandom testing environment is a challenging task. In this paper, we propose a new BIST design that prevents the LFSR from cycling through the states generating useless patterns, as well as reorders the useful test patterns in a desired sequence to minimize total energy demand.

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 **IEEE**

IEEE COMPUTER SOCIETY

To estimate energy loss, we compute the total SA as the number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions in all the circuit nodes including the LFSR, CUT, and the scan path over a complete test session. The various steps of the proposed method are now summarized below:

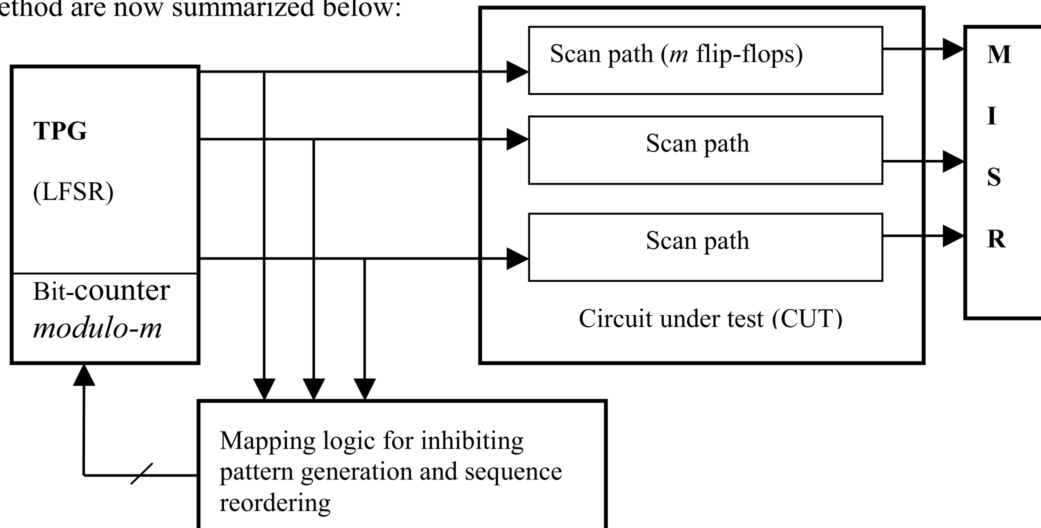

Fig. 1: Proposed low-energy BIST design

(i) A pseudorandom test sequence is generated by an LFSR, and its single stuck-at fault coverage in the CUT is determined through forward and reverse fault simulation; let S denote the test sequence up to the last useful vector (beyond which fault coverage does not improve significantly).

(ii) Identify the set U of useless patterns in S that do not contribute to fault dropping.

(iii) For all ordered pairs of test vectors in the reduced set $S_r = (S \setminus U)$, determine the switching activity (SA) in the scan path and the CUT.

(iv) Reorder the vectors in S_r to estimate an optimal order S' to minimize energy.

(v) Modify the state table of the LFSR such that it generates the new sequence S'.

(vi) Synthesize a mapping logic (ML) with minimum cost, to augment the LFSR; the state transitions of the LFSR are modified under *certain conditions* to serve two purposes: (a) to prevent it from cycling through the states generating useless patterns, and (b) to reorder S_r to S'; for all other conditions, the LFSR runs in accordance to its original state transition function. Fig. 2 shows a simple MUX-based design that can be used for this purpose. A similar idea of skipping LFSR states is used earlier for embedding a set of deterministic tests [10].

The following example of a TPG (see Fig. 3; ignoring the dotted arcs) illustrates the idea of state-skipping technique. The LSB of the LFSR is shifted serially into the scan path generating a test sequence S (Table 1). Some component of SA is *intrinsic* (invariant over a full test session), and the rest is *variable*. Hence, SA can be represented as a directed complete graph called activity graph (Fig. 4b), where each node represents a test vector, and the directed edge (e_{ij}) represents application of the ordered test pair (t_i, t_j). The weight $w(e_{ij})$ on the edge e_{ij} denotes the variable component of SA corresponding to the ordered pair of tests (t_i, t_j). The intrinsic component being independent of test ordering is represented as a node weight and may be ignored as far as the optimal ordering is concerned. An example of an activity graph is shown in Fig. 4b. The edge weights are represented as an asymmetric cost matrix (Fig. 4a), as the variable component of SA strongly depends on ordering of test pairs. Thus, for the test sequence S ($t1 \rightarrow t2 \rightarrow t3 \rightarrow t4 \rightarrow t5$), the variable component of switching activity is 37. Now, if t3 is found to be a useless test pattern, it along with all incident edges, can be deleted. An optimal ordering of test vectors that minimizes the energy consumption is a min-cost Hamiltonian path: S' ($t1 \rightarrow t2 \rightarrow t5 \rightarrow t4$), the path cost being equal to 23 (Fig. 4c).

IEEE COMPUTER SOCIETY

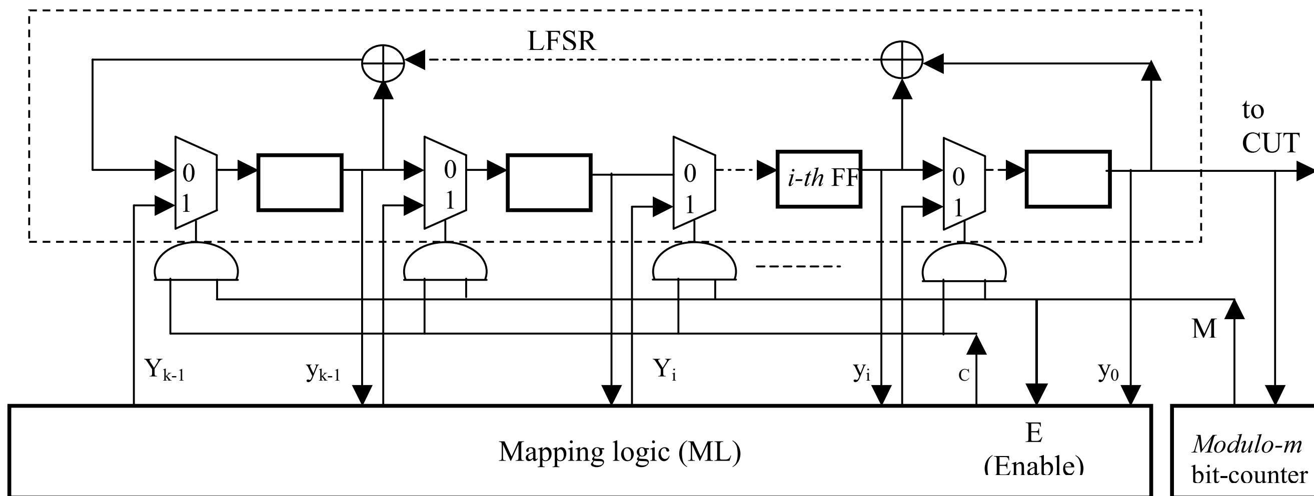

Fig. 2: Modification to the LFSR

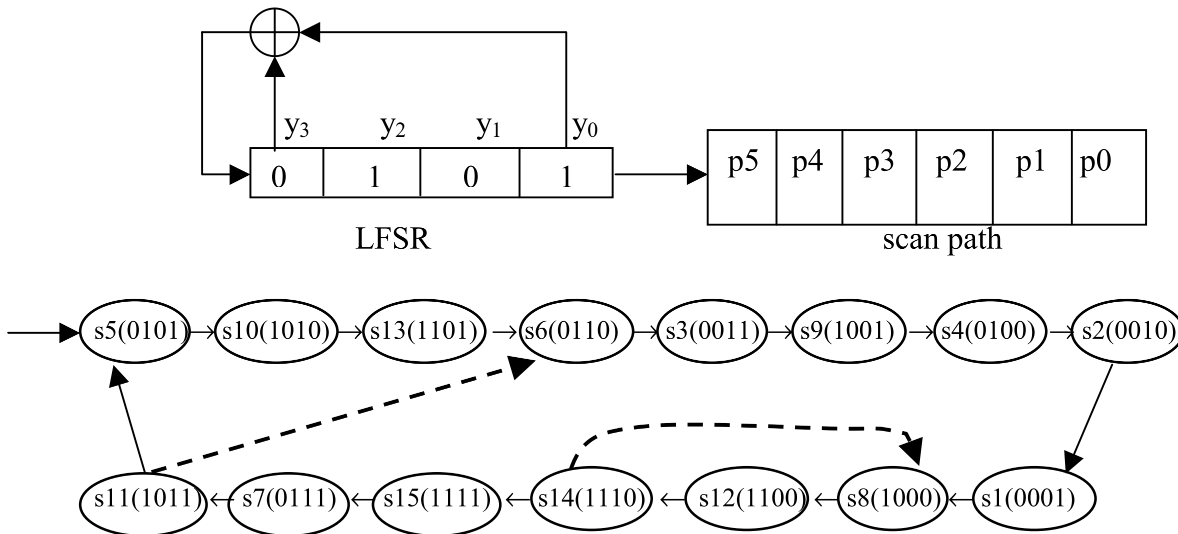

Fig. 3: An example LFSR and its state diagram

Table 1: Test vectors and their start- and end-states

Test sequence S	p0	p1	p2	p3	p4	p5	start-state LFSR	end-state LFSR
t1	1	0	1	0	1	1	s5	s9
t2	0	0	1	0	0	0	s4	s14
t3	1	1	1	1	0	1	s15	s13
t4	0	1	1	0	0	1	s6	s1
t5	0	0	0	1	1	1	s8	s11

Thus, in the new sequence *S′,* for the ordered pair (t1 → t2), no action is required, as t2 is generated by the LFSR as a natural successor of t1. So, for s9 (end-state of t1), we set the Y-outputs of the mapping logic (see Fig. 2) to don't cares (*d*), and the control line C to 0. However, we need an additional transition from s14 (end-state of t2) to s8 (start-state of t5), and similarly from s11 (end-state of t5) to s6 (start-state of t4). For these combinations, the Y-outputs are determined by the corresponding start-states, and C is set to 1. For all other remaining combinations, all outputs are don't cares. These transitions generate the useful test patterns in a desired sequence, and prevent the LFSR to cycle through the states that generate useless patterns (in this example, test t3). Further, the output M of the *modulo-m* bit counter assumes 1 only when scan path (whose length is *m*) is filled, i.e., at the end-states of

IEEE COMPUTER SOCIETY

the test vectors. Thus, in order to generate the sequence *S′*, we need to skip the natural next state of the LFSR and jump to the start state of the desired next test pattern. These state skipping transitions are shown with dotted lines in Fig. 3.

	t1	t2	t3	t4	t5
t1	0	8	9	11	9
t2	11	0	10	12	7
t3	8	6	0	10	6
t4	10	9	6	0	9
t5	11	8	7	8	0

Fig. 4a: Cost matrix

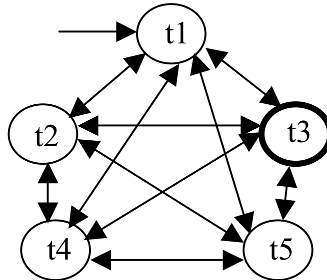

Fig. 4b: Activity graph

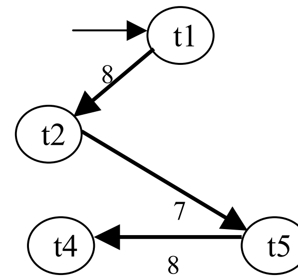

Fig. 4c: Optimal H-path

In general, the mapping logic can be described as follows: given a seed, let *S* denote the original test sequence generated by the LFSR, and $S' = \{t'_1, t'_2, \ldots t'_i, t'_{i+1}, \ldots\}$ denote the optimally ordered reduced test sequence consisting of useful vectors only. Let y_i denote the output of the i-th flip-flop of the LFSR, and Y_i denote the output of the mapping logic feeding the i-th flip-flop through a MUX (see Fig. 2). The ML is a combinational circuit with k inputs $\{y_0, y_1, \ldots, y_{k-1}\}$, and k+1 outputs $\{Y_0, Y_1, .., Y_{k-1}, C\}$, where k is the length of the LFSR, and C is a control output. For every test t'_i in *S′*, there is a corresponding row in the truth table given by (Table 2):

Table 2: General description of the mapping logic

	Present State of the LFSR	Outputs of the mapping logic	
	y_{k-1} ... y_0	Y_{k-1} ... Y_0	C
Case (i)	End state of t'_i	*d d d d*	0
Case (ii)	End state of t'_i	start-state(t'_{i+1})	1
	all other states	*d d d d*	*d*

Case(i) is applicable if the consecutive test pair (t'_i, t'_{i+1}) of *S′* appears in consecutive order in the original sequence *S* as well; otherwise, *case (ii)* is applicable. Thus, the next-state of the LFSR follows the transition diagram of the original LFSR when either C = 0, or M = 0, and is determined by the outputs of the mapping logic if and only if CM = 1. Since these additional transitions emanate only from the end-states of test patterns, their occurrences can be signaled by the M output, and also when C = 1. In order to prevent the SA from occurring in ML for every scan shift cycle, an enable signal E controlled by M is used. Thus, the y-inputs become visible to ML if and only if M = 1. The test session terminates when the end-state of the last useful pattern in *S′* is reached. Determination of optimal reordering of test patterns is equivalent to solving a traveling salesman problem (TSP), which being NP-hard, needs heuristic techniques for quick solution [9].

IV. Experimental results

We assume fully isolated scan-path architecture [12], where the scan register (SCAN) is completely separated from the CUT by a buffer register (BUFFER). This eliminates SA rippling through the CUT while shifting in a test pattern. When all the bits of a test pattern are shifted in, the content of SCAN is copied to BUFFER, and then applied to the CUT. During the system mode, the CUT outputs are captured in BUFFER, and then copied to SCAN. The response vector is shifted out while shifting in the next test pattern to SCAN. A *test cycle* corresponds to applying a particular test vector to the CUT and capturing its response. A major component of scan-shift SA although being dependent on consecutive vector pairs, remains invariant over a complete test session. Hence, this portion may also be treated as intrinsic as far as TSP optimization is concerned.

Experiments were carried out on several ISCAS-89 scan benchmark circuits, and results are reported in Tables 3 through 6. A 25-bit LFSR is used to generate 20,000 pseudorandom test vectors, and the useless vectors are identified and eliminated by running the HOPE fault simulator [7]. Since the modified LFSR generates only the useful patterns, a significant amount of test application time is saved as

IEEE COMPUTER SOCIETY

shown in the last column of Table 3. Next, for each pair of useful test vectors, SA in the CUT and the scan path is computed. We assume a single linear scan chain. Table 5 shows the (intrinsic) SA due to scan shift and capturing the responses in the BUFFER. To determine an optimum reordering of useful test vectors, we consider only the variable component of SA occurring in the CUT and the scan path for every ordered pair of useful vectors. We then run a TSP solver [8, 9] to find a nearly optimal ordering. Next, the mapping logic (ML) for the LFSR is synthesized using ESPRESSO [15] and SIS [16].

Table 4 indicates a significant amount of total energy savings in the LFSR. For small-sized circuits, the relative overhead of ML is high compared to the cost of the CUT, as we have used a 25-bit LFSR. However, for large circuits, e.g., s35932, the overhead of ML is low. Table 6 depicts the reduction of order-dependent (variable) component of SA by optimal ordering of useful patterns. The CPU time on SUN Ultra 10 (233 MHz) for solving the TSP is also reported. Table 5 depicts a pessimistic observation that a major component of SA in the scan path is *intrinsic* in nature, which cannot be reduced by test reordering.

V. Conclusion and future problems

A new BIST design is described for saving energy both in the LFSR and the CUT in a random testing environment. A significant component of the SA is observed to be intrinsic in nature, which given a test set, cannot be reduced by vector reordering. To reduce this component, either a different set of useful test vectors is to be selected from the random

Table 3: Useful patterns and savings in test application time

Circuit	Number of FF's	Number of PI's	Number of PO's	Number of useful patterns	Last useful pattern position	Fault coverage	Savings in test application time (%)
s208.scan	8	11	2	31	2992	100.000	98.96
s953.scan	29	16	23	103	18306	99.907	99.44
s1423.scan	74	17	5	75	19449	98.878	99.61
s5378.scan	179	35	49	267	19950	98.682	98.66
s9234.1.scan	211	36	39	290	19567	85.347	98.52
s15850.1.scan	534	77	150	307	19984	91.616	98.46
s35932.scan	1728	35	320	71	197	89.809	63.96
s38417.scan	1636	28	106	620	19879	94.403	96.88
s38584.1.scan	1426	38	304	719	19885	95.160	96.38

Table 4: Energy savings in the LFSR

1 Circuit	2 SA in the original LFSR	3 SA in the modified LFSR	4 Cost of mapping logic (# literals)	5 SA in the mapping logic (ML)	6 Total SA in the modified LFSR (3+5)	7 Energy savings (%) (2-6)/2
s208.scan	784901	7763	342	1937	9700	98.76
s953.scan	10852686	52480	1100	9426	61906	99.43
s1423.scan	18304013	77295	779	5718	83013	99.55
s5378.scan	47336298	733053	3161	48422	781475	98.35
s9234.1.scan	61658727	896855	3292	52314	949169	98.46
s15850.1.scan	149195549	2533658	3799	65599	2599257	98.26
s35932.scan	6235825	1594065	841	6523	1600588	74.33
s38417.scan	41575505	12348460	7301	152168	12500628	69.93
s38584.1.scan	360243110	12719030	8580	187308	12906338	96.42

IEEE COMPUTER SOCIETY

sequence, or the scan path architecture is to be radically redesigned. Another intrinsic source of power consumption is the clocking circuitry, which is not considered in this work. Ensuring reusability of mapping logic and BIST hardware for different cores on a chip is another open area to study.

Table 5: Intrinsic SA due to useful patterns

Circuit	Scan path	Capture	Total SA
s208	3599	998	4597
s953	110958	10489	121447
s1423	246292	18631	264923
s5378	5683786	299625	5983411
s9234.1	8884279	631840	9516119
s15850.1	75741475	1362104	77103579
s35932.	140635082	569135	141204217
s38417.	870769045	5770204	876539249
s38584.1	1021004396	5403604	1026408000

References

[1]. S. Gerstendoerfer and H. -J. Wunderlich, Minimized power consumption for scan-based BIST, *Proc. ITC,* pp. 77-84, 1999; (also in *JETTA*, January 2000).

[2]. S. Wang and S. K. Gupta, DS-LFSR: A new BIST TPG for low heat dissipation, *Proc. ITC,* pp. 848-857, 1997.

[3]. R. M. Chou, K. K. Saluja, and V. D. Agrawal, Scheduling tests for VLSI systems under power constraints, *IEEE TVLSI ,* pp. 175-185, June 1997.

[4]. D. Gizopoulos, et al., Low power/energy BIST scheme for datapaths, *Proc. VTS,* pp. 23-28, 2000.

[5]. R. Sankaralingam, et al., Static compaction techniques to control scan vector power dissipation, *Proc. VTS*, pp. 35-40, 2000.

[6]. V. P. Dabholkar, et al., Techniques for minimizing power dissipation in scan and combinational circuits during test application, *IEEE TCAD,* vol. 17, pp. 1325-1333, Dec. 1998.

[7]. H. K. Lee and D. S. Ha, HOPE: An efficient parallel fault simulator for synchronous sequential circuits, *IEEE TCAD*, vol. 15, pp. 1048-1058, Sept. 1996.

[8]. http://lancet.mit.edu/galib-2.4/

[9]. D. S. Johnson, et al., Experimental analysis of heuristics for the ATSP, in *The Traveling Salesman Problem and its Variations,* Gutin and Punnen *(Ed.),* Kluwer Academic Publisher, 2002.

[10]. S. J. Upadhyaya and L-C. Chen, On-chip test generation for combinational circuits by LFSR modification, *Proc. ICCAD*, pp. 84 – 87, 1993.

[11]. N. Nicolici, B. M. Al-Hashimi , Multiple scan chains for power minimization during test application in sequential circuits, IEEE TC, pp. 721-734, June 2002.

[12]. M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. IEEE Press, 1999.

[13]. S. Wang and S. K. Gupta, LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation, *Proc. ITC*, pp. 85-94, 1999.

[14]. A. Chandra and K. Chakrabarty, Low-power scan testing and test data compression for system-on-a-chip, IEEE Trans. CAD, pp. 597-604, May 2002.

[15]. R. K. Brayton, et al., *Logic Minimization Algorithms for VLSI Synthesis.* Kluwer, Boston, 1984.

[16]. E. M. Sentovich, et al., SIS: A system for sequential circuit synthesis, *Tech. Rep. UCB/ERL M92/41*, Electronic Research Lab., 1992.

[17]. Y. Zorian, A distributed BIST control scheme for complex VLSI devices, *Proc. VTS,* pp. 4-9, 1993.

Table 6: Reduction of order-dependent component of SA by test reordering

	SA in the natural LFSR order			SA in the optimal TSP order			Energy savings (%)			CPU
Circuit	Scan path	CUT	Total	Scan path.	CUT	Total	Scan path	CUT	Tot.	time(sec.)
s208	958	958	1916	114	759	873	88.1%	20.8%	54.4%	90
s953	2704	10198	12902	2652	8162	10814	1.9%	20.0%	16.2%	255
s1423	2821	19689	22510	546	16739	17285	80.6%	15.0%	23.2%	169
s5378	30388	298565	328953	4280	276382	280662	85.9%	7.4%	14.7%	1283
s9234.1	38750	646794	685544	39000	559996	598996	-0.6%	13.4%	12.6%	1418
s15850.1	110124	1338785	1448909	110124	1208780	1318904	0.0%	9.7%	9.0%	1912
s35932	67584	582776	650360	67584	503881	571465	0.0%	13.5%	12.1%	181
s38417	498212	5745334	6243546	496470	5521566	6018036	0.3%	3.9%	3.6%	5611
s38584.1	598580	5389292	5987872	600310	5169447	5769757	-0.3%	4.1%	3.6%	7803

1063-9667/03 $17.00 © 2003 **IEEE**

IEEE COMPUTER SOCIETY