A New Low Energy BIST Using A Statistical Code

Sunghoon Chun, Taejin Kim and Sungho Kang
Department of Electrical and Electronic Engineering Yonsei University
134 Shinchon-dong Seodaemoon-gu, Seoul, Korea
Tel: : +82-2-2123-2775, Fax : +82-2-313-8053
E-mail: <shchun, tj2010, shkang>@yonsei.ac.kr

Abstract - To tackle with the increased switching activity during the test operation, this paper proposes a new built-in self test (BIST) scheme for low energy testing that uses a statistical code and a new technique to skip unnecessary test sequences. From a general point of view, the goal of this technique is to minimize the total power consumption during a test and to allow the at-speed test in order to achieve high fault coverage. The effectiveness of the proposed low energy BIST scheme was validated on a set of ISCAS '89 benchmark circuits with respect to test data volume and energy saving.

I. Introduction

As VLSI circuits are growing increasingly more complex, exact testing of VLSI circuits is gaining its importance. Since today's large and complex VLSI circuits in SoC environments need an enormous amount of test data, their exact testing becomes even more difficult. When SoCs are tested, the test data are transferred to the circuit under test (CUT) from automatic test equipment (ATE). Since the channel width and the size of memory for the ATE are limited, the traditional ATE must either be modified or replaced with a more expensive ATE to test an SoC with enormous test data. In addition, if the original test data are reduced for the size of the ATE memory by eliminating useful test patterns, the accuracy of testing is diminished. Thus, test data compression is essential in overcoming these limitations and researchers have been presenting various built-in-self-test (BIST) schemes to alleviate these problems.

In addition to the problem of test data volumes, the test power and the energy consumption has become another major problem for a SoC test. The switching activities during the test mode could be twice as high as those of the normal mode [1] and excessive energy consumption during testing caused by the excessive switching activities can cause several problems.

Therefore, we focus on a BIST scheme to reduce the energy required for test in order to overcome these problems. In fact, the low energy scan testing and the reduced test data volume are conflicting goals since large test data and more effective test data guarantee higher fault coverage while they increase the power consumption during testing. To alleviate these conflicting goals, many approaches to resolve the test problems have been researched over the last several years [2-10]. However, these techniques lead to high area overhead and the large test data volume and hence are not practical for large circuits.

This paper proposes a new pattern generator based on a new compression code with the modified input reduction scheme and a new test sequence skipping technique. The proposed pattern generator consists of three main parts: 1) a modified input reduction scheme to more efficiently reduce test data and test power consumption; 2) a statistical and simple compression code to generate more effective and smaller test sets; 3) a new test sequence skipping technique for significantly reducing wasted power consumptions with a lower hardware overhead during the self-testing. From a general point of view, the goal of this technique is to minimize the total power consumption during a test and to allow the at-speed test in order to achieve high fault coverage.

This paper is organized as follows. The next section explains the energy consumption model to estimate the total energy consumption of our method. In Sections 3 and 4, a new pattern generator for a low energy test in BIST-based designs is proposed and experimental results for it are reported. The conclusion is given in Section 5.

II. Energy Consumption Model

It has been shown [11, 12] that during the normal operation of well-designed CMOS circuits the dynamic energy dissipation caused by the switching activity accounts for over 90% of the total energy consumption. Thus, total energy optimization techniques have been employed at different levels of abstraction target minimal switching activity. The model for energy consumption for the gate i in a logic circuit is simplified as:

$$E_i = \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot N_i \qquad (1)$$

The capacitive load that the gate is driving, C_i, can be extracted from the circuit. The switch activity N_i depends on the input to the system which during the test mode involves test vectors and therefore, for low energy scan based tests, reducing switching activities during scan shifting is one of the most significant factors.

To find the BIST schemes to minimize the energy consumption by switching activities during scan shifting, a means for comparing the energy consumption by two vectors is needed. The most accurate results would be obtained by using a circuit simulator that finds the actual number of circuit elements that switch when a vector is scanned in and out. Since this process of using a simulator is costly because of execution time, a simple heuristic is required for comparing the power and the energy dissipated by two vectors. Such a heuristic is presented in [13].

In this paper, the extended version of the weighted

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

transition metric (WTM) introduced in [13] is used to estimate the energy consumption caused by the scan vectors. In the original WTM, the scan-in energy consumption for a given vector is estimated using the number of transitions in it with their relative positions. However, since the scan-out energy consumption for a given vector should be considered during a scan based test, we propose the extended transition metric (ETM).

To have the energy consumption of the scan out vectors, the CUT must be simulated. However, since we focus on reducing the energy dissipation for the pattern generator and scan paths during the BIST mode, we do not consider the power dissipation in the CUT.

Consider a scan chain of length k, a test vector P = {p_1, p_2, ……, p_i} and a scan response R = {r_1, r_2, …… , r_i}. The ETM to estimate the energy consumption during the scan based test is as follows:

$$ETM = \sum_{j=1}^{i-1} (p_j \oplus p_{j+1}) \cdot (k-j) + \sum_{j=1}^{i-1} (r_j \oplus r_{j+1}) \cdot (k-j) \qquad (2)$$

If the total test vector set P_{total} contains n vectors p_1, p_2, …… , p_n and n response r_1, r_2, ……, r_n, then the total energy consumption ETM_{total} would be

$$ETM_{total} = \sum_{i=1}^{n} ETM_i \qquad (3)$$

III. The Proposed Low Energy BIST

The proposed low energy BIST scheme has three main phases; First phase is to prepare an initial test set, second is to generate a pattern generator using a statistical code and a skipping logic for low energy test is generated as the final phase. Fig.1 shows the overall algorithm of the low energy BIST generation.

```
Low_Energy_BIST()
{
  /* Phase 1: preparing an initial test set */

     T = generate_pseudo-random_pattern();
     T' = don't_care_identification();
     T'IR = input_reduction();

  /* Phase 2: generating a pattern generator using the MICRO code */

     determine_the_compression_block();
     generate_the_weighted_logic();
     generate_the_decompressor_for_the_MICRO();
     generate_pattern_generator_using_the_MICRO_code();

  /* Phase 3: generating a skipping logic for low energy test */

     determine_success_stages();
     reorder_the_success_stages();
     calculate_the_seeds_for_the_success_stages();
     synthesize_the_skipping_logic();

}
```

Figure 1. Overall algorithm of the proposed low energy BIST generation

A. Preparing An Initial Test Set

Initially, a pseudo-random pattern of the LFSR is generated as the original test set T. Since the traditional input reduction [14] uses a method of analyzing the relation of inputs in a given netlist, the approach in [14] is not suitable for modern large circuits.

We proposed the modified input reduction in [15] to overcome this limitation. Unlike the input reduction in [14], we considered a case where test sets are given by the deterministic ATPG or the LFSR. Since the input reduction approach uses deterministic test patterns, it is not perfectly adapted to the actual BIST logics. To increase the accuracy of the input reduction by using the test patterns, the pseudo-random test patterns which are actually generated from the LFSR in BIST logics can be used. In this case, many specified bits in pseudo-random patterns have no influence in activating and propagating faults. In addition, they make it difficult to identify the compatible inputs and inverse compatible inputs. The process of don't care bit identification [16] is required for a given test set obtained from the LFSR in order to reduce test inputs efficiently.

After the don't care identification procedure, compatible inputs for the reassigned test set are computed by using the input reduction algorithm in [15]. Note that the test set after the modified input reduction denotes T'_{IR}.

B. Generating a new pattern generator

This section describes a new pattern generator that uses a statistical code. The new pattern generator is proposed in order to efficiently compress test data for low energy test.

In recent work published in [17], it was shown that the Huffman code is close to entropy limits and it can achieve the highest compression ratio among other statistical codes. However, for the Huffman code, as the number of the compression blocks is increased, the hardware overhead for the decompressor is increased. In general, since high hardware overhead leads to large power and energy consumptions, we select the MICRO code in [15] to appropriately satisfy both high compression ratio and low hardware overhead among many statistical compression code. The idea of the MICRO code [15] is that the compression ratio is enhanced by increasing the occurrence frequency of one block. The detail procedures of determining the compression block and generating the MICRO code word are explained in [15].

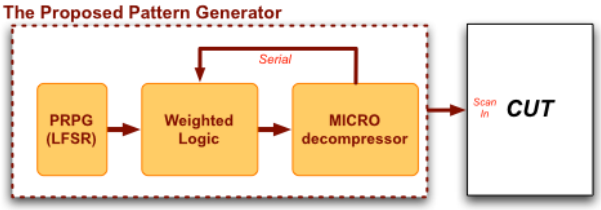

Figure 2. The architecture of the proposed pattern generator

The overall architecture of the proposed pattern generator is shown in Fig. 2. The proposed pattern generator consists of a general LFSR, a weighted logic and the MICRO decompression logic. The weighted logic is designed to make the compression block occur as much as the probability of the occurrence frequency of one block in the T_{IR}' or T'. Note that if it is possible to change the circuit

design or scan chain organization, the T_{IR}' is used to reduce more test data volume and more test application time.

When the proposed pattern generator just used the MICRO code, it was able to determine the combination of the compression code and a general LFSR. Of course, the combination of both is advantageous by itself. However, in this paper, we added a weighted logic to the combination of the compression code and a general LFSR in order to reduce the total test energy and power consumption and to achieve high fault coverage with a smaller number of patterns than the conventional BIST architecture for low energy testing. One of the possible solutions to increase the fault coverage of a general LFSR is to use weighted random patterns [18]. Traditional weighted random patterns increase the fault coverage by using the signal probability of the CUT. Similarly, we can increase the fault coverage by using the weighted logic for the MICRO code. For the MICRO code, a pseudo-random pattern test will be more efficient if it will generate more compression blocks than the uncompression blocks. For this reason, we used a weighted logic for the proposed pattern generator to reduce the total test energy and to achieve high fault coverage with a smaller number of patterns.

During the determining process of the compression block(*CB*), the occurrence frequency of the *CB* can be stored. The stored occurrence frequency of the *CB* is used as the weight between the decompressor of the MICRO code and an LFSR. The proposed weight logic implements with the probability of generating the *CB* one-bit code at the input of the MICRO decompressor. This pattern generator effectively deals with hard-to-detect faults by generating the *CB* code word as the probability of the *CB* occurrence frequency. In the conventional LFSR, each scan input has a probability of 0.5 of being either a 0 or a 1. In the proposed weighted LFSR, the probability is adjusted so that the code word for the compression block is more likely generated as the stored *CB* occurrence frequency. If each LFSR stage has a probability of 0.5 of being either 0 or 1 and is statistically independent of the values on the other LFSR bits, then ANDing k LFSR signals would result in a 1 value at the AND gate output with a probability of 0.5^k. On the contrary, ORing k LFSR signals result in a 0 value at the OR gate output with a probability of 0.5^k. INVERTERs can be used to obtain other probabilities. Therefore, the probability of the *CB* uses a value of 0.5^k in order to implement the weight logic simply. In addition, the masking logic is required to mask the weighted logic during the uncompression block decoding stage in the MICRO decompressor because each LFSR stage has a probability of 0.5 for the *UB*s (uncompression blocks).

The FSM decoder of the MICRO code loads the compression block *CB* in parallel with the *Parallel* output signal into the controller. If the prefix bit is a '1', then the FSM decoder simply transfers the next two bits with the *Serial* output signal into the controller during the next two cycles since the prefix bit '1' indicates that the next bits are not encoded. Therefore, if the value of the *Serial* output signal is 1, the masking logic is activated to generate an encoded test pattern as the original probability. This masking logic is easily implemented with the *Serial* signal and one MUX. Fig. 3 shows an example of the proposed LFSR with a weighted logic and a masking logic when the probability of the *CB* occurrence is 0.625 and the 4-bit LFSR is used.

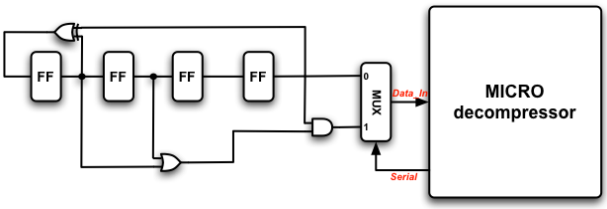

Figure 3. An example of the proposed pattern generator

The proposed pattern generator can easily resolve the linear dependency problem of a general LFSR. The recurrence relation of a general LFSR causes linear dependencies within the sequence that are of importance to their use as test stimuli. In general, the test for a particular fault in the CUT requires that the k inputs to that circuit take on certain specified values. In a conventional LFSR, some test patterns for particular faults cannot be generated because of the effect of the linear dependency problem when the span of the sampling polynomial exceeds the length of the LFSR that generates the sequence. However, the proposed pattern generator has more chances to generate tests for hard-to-detect faults than a general LFSR even though the span of the sampling polynomial occurs because the sequences of the proposed pattern generator do not have the recurrence relation caused by the influences of the weighted logic and the MICRO code. Therefore, the fault coverage of the proposed pattern generator with a smaller number of patterns can be much higher.

Furthermore, the proposed pattern generator requires an LFSR with fewer states than a general LFSR on account of the MICRO code and the weighted logics. The number of states that a general LFSR can cycle through is less than or equal to 2^n-1, where n is the number of stages in the general LFSR. Therefore, to test large circuits which include many test inputs, the number of stages in the general LFSR should be numerous. However, the proposed pattern generator generates upwards of 2^n-1 stages since a period of the *CB* code occurrence is changed according to the weighted logic. Hence, it reduces the number of stages in the LFSR with the consequence that the hardware overhead and the energy and power consumption for the LFSR in the proposed pattern generator are diminished.

C. Generating A Skipping Logic

In a general BIST architecture, a modulo-m bit counter keeps track of the number of scan shifts, where m is the length of the longest scan paths. As shown in Fig. 4, since the number of useful patterns is known to be a very small fraction of all the patterns generated, 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. Through fault simulation, it can be easily verified that not all patterns in a pseudo-random test set can detect faults in the circuit. A test set which is generated from the proposed pattern generator also includes non-detecting subsequences of test patterns. Therefore, a new BIST design that prevents the pattern generator from generating useless patterns is required to minimize the total test power consumption with low hardware overhead.

Figure 4. Useless patterns of a general LFSR

The proposed BIST architecture is shown in Fig. 5. This sub-section describes a new test vector skipping technique in the proposed BIST architecture. This skipping technique is a little similar to the inhibiting technique [2]. The inhibiting technique used the decoding logics and transmission gates networks to skip the inhibiting test sequences. However, its test energy reduction ratio is still low and its hardware overhead can be much higher for multiple LFSR inhibitions. Therefore, to achieve higher energy reduction ratio, we propose a new skipping logic technique using the MICRO code. In addition, a new success stage reordering algorithm is proposed to find an optimal test sequence with a low hardware overhead. Since the proposed skipping logic includes the logic for generating seeds, this technique can be easily extended to the reseeding technique [19] to deal with hard-to-detect circuits.

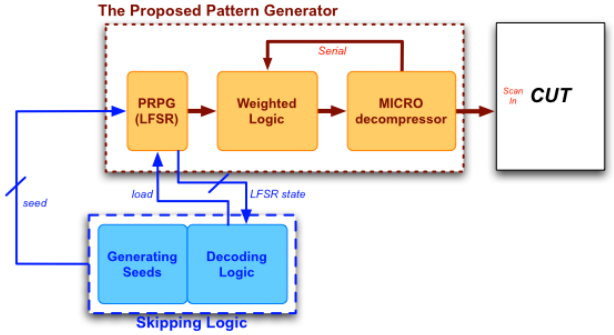

Figure 5. Low energy BIST based on the skipping technique

The procedure of developing the skipping structure consists of three phases. The first phase is to generate a pseudo-random test sequence by the proposed pattern generator and to determine its single stuck-at fault coverage in the CUT through fault simulation. Then, the first and last vectors of each useless subsequence in the whole sequence generated by the proposed pattern generator are identified, such as V_1 and V_2 in Fig. 4. We call each useful subsequence the success stage S, such as S_1 and S_2 in Fig. 4.

For low power and low energy testing, modification logic/ROM may be used to inhibit or skip the LFSR states that generate useless test patterns. The extra logic overhead increases rapidly with the number of such jumps. Therefore, we propose a new success stage reordering algorithm to find an optimal test sequence with a low hardware overhead. In the second phase for finding an optimal test sequence, the number of transitions between each success stages is calculated and the number of faults detected in each success stage is calculated through fault simulation. Then, a state transition graph $G(V,E)$, where V is a success stage S_i and E is a skipping constraint value, is generated to determine the optimal sequence of success stages. Note that the formulation of the skipping constraint value consists of a number of transitions between success stages (N_{Tr}) and a number of detected faults of V after a predecessor (N_{DF}). The skipping constraint value (SCV) is calculated as follows.

$$SCV = \alpha \cdot N_{DF} + \beta \cdot N_{Tr} \qquad (4)$$

where α and β are weighted constants of N_{DF} and N_{Tr}, respectively and they are all user defined values. To minimize the skipping logic, the minimum success stages with minimal transitions are required. Therefore, it is important to determine the appropriately weighed constants. We assume that α and β are 1 in this paper.

The goal of this step is solving the problem of minimizing the SCV in order to reduce the area and the energy consumption of the skipping logics. We propose the following approach to find an optimal solution.

a) For a given stage transition graph, choose the success stage S_i which has the maximum fault coverage.
b) S_i sets the start stage.
c) Select the next success stage S_j which has the minimum SCV_{ij} between S_i and S_j.
d) Drop the success stage S_i in a given stage transition graph and re-calculate the N_{DF}s and N_{TR}s for new skipping constraint values.
e) Create a new stage transition graph and then S_j set the start stage.
f) Repeat from the step c) to f) until that the fault coverage is 100% or only one vertex in a stage transition graph exists.

The proposed success stage reordering algorithm is capable of minimizing the area and the energy consumption of the skipping logic. In addition, the proposed algorithm was further extended to use a reseeding technique to deal with hard-to-test circuits or reduce test application time and test data volume more. The reseeding method is a technique in which the LFSR used to generate pseudo-random patterns is loaded with different seeds from which it produces vectors that cover the test cubes of random pattern resistant faults. Therefore, some seeds can be calculated from the test cubes of the CUT to use the reseeding technique and then each calculated seed is regarded as one of the success stages. Thus, the proposed algorithm can be easily extended to find the optimal useful pattern subsequences by including the success stages generated by seeds into the state transition graph.

In the third phase, the skipping logic with the minimum cost is synthesized to prevent unnecessary pattern subsequences and to jump the first vectors of success stages as the order obtained in the second phase. The skipping logic

consists of a decoding logic, which is connected to the output of the LFSR, and the state of the next useful patterns. To generate the start state of the next success stage, the first pattern of the success stage is reversely encoded by using the MICRO code, and then the LFSR state is calculated by solving the linear equation for the reversely encoded pattern. The seed for the LFSR start state of the first pattern of useful subsequence is easily calculated by the Gauss-Jordan elimination. However, the number of skipping states should be limited in order to reduce the area overhead for the skipping logic.

IV. Experimental Results

To demonstrate the efficiency of the proposed method, the proposed low energy BIST is used for ISCAS '89 benchmark circuits. The proposed method and previous approaches [2, 7, 10] were implemented in C and the experiments were performed on a Pentium 4 1.4GHz system with Linux. The in-house ATPG and the fault simulator were used to calculate the number of vectors to achieve definite fault coverage for a general BIST scheme. In Table 1, the number of patterns of every general BIST session and the total fault coverage are presented. In Table 1, column 2 is the number of LFSR bits for a general LFSR and previous studies, and column 3 is the number of vectors generated by a general LFSR to achieve appropriate fault coverage. Note that the polynomial equation of the applied LFSR with 10 scan chains for each circuit is generated by using the commercial tool, the LBISTArchitect [20] in Mentor Graphics with the TSMC 0.25μm library.

TABLE 1. INITIAL DATA USED FOR EXPERIMENTS

Circuits	Required LFSR bits	Number of Patterns	Fault Coverage (F.C) (%)
s5378	22	50000	84.39
s9234	22	50000	96.42
s13207	24	50000	97.78
s15850	23	50000	93.61
s38417	25	70000	98.25
s38584	25	70000	98.90

To present the efficiency of the proposed pattern generator, its test patterns were generated in each circuit until the similar or higher fault coverage was achieved. In this case, the number of generated patterns depends on the number of skipping non-detecting sequences. Since the number of subsequences to be skipped is related to the hardware overhead, a large number of them is not always the optimal solution. For the proposed pattern generator, experiments were carried out on the same ISCAS '89 benchmark circuits as in Table 1, and the results of the previous works and the proposed method for test data with the input reduction scheme and without the input reduction scheme are recorded in Table 2 and Table 3, respectively. For each circuit, using the in-house tool, we calculated the required polynomial equation of the applied LFSR for the same or higher fault coverage than that of Table 1. Note that the reduction ratio is computed as follows.

$$reduction\ \ ratio\ = \frac{N_{LFSR} - N_{Proposed}}{N_{LFSR}} \times 100 \qquad (5)$$

where the N_{LFSR} is the number of patterns for a general LFSR and the $N_{Proposed}$ is the number of patterns for the proposed pattern generator.

In Table 2, the results of [2] were obtained by using a multiple LFSR inhibition scheme and the results of [10] were obtained when the gap value was 10. As shown in column 2 in Table 3, the fault coverage is higher than that of previous studies.

TABLE 2. THE TEST DATA OF PREVIOUS WORKS

Circuits	General LFSR	[2]	[7]	[10]
	Test data (bits)	Test data (bits)	Test data (bits)	Test data (bits)
s5378	10,700,000	5,485,248	1,376,448	537,140
s9234	12,350,000	7,390,240	2,726,880	679,250
s13207	35,000,000	19,801,600	10,035,200	2,940,000
s15850	30,550,000	18,124,704	6,041,568	1,857,440
s38417	116,480,000	34,305,408	21,305,856	9,967,360
s38584	102,480,000	50,654,400	26,351,360	9,311,040

The longer test sequence and the greater volume of test data for a CUT mean that the energy consumption to test the CUT can be increased and thus the probability of damaging the circuit during a test session can be much higher. In addition, since the proposed pattern generator requires smaller bits for LFSR on account of the MICRO code, the energy dissipation to test a CUT was further reduced. Experimental results were obtained in this manner, and the results of energy consumption and the area overhead are presented in Table 4 and Table 5. The energy consumptions and the area overheads for the proposed BIST in Table 5 were calculated in the cases for high data compression in Table 3. For each test pattern, ETM in the scan path was computed. As can be seen, these results reveal a significant reduction in ETM compared with the conventional

TABLE 3. THE TEST DATA OF THE PROPOSED METHOD

Circuits	F.C of [2], [7] and [10]	Proposed								
		F.C	For low hardware overhead				For high data compression			
			Test data (w/o IR) (bits)	Test data (w/ IR) (bits)	# of skips	Reduction ratio	Test data (w/o IR) (bits)	Test data (w/ IR) (bits)	# of skips	Reduction ratio
s5378	84.4	99.0	471,014	407,296	33	96.2	209,934	**181,632**	47	**98.3**
s9234	96.4	97.3	941,811	671,232	95	94.6	327,522	**233,472**	127	**98.1**
s13207	97.8	95.8	4,767,000	3,677,440	67	89.5	2,086,700	**1,610,240**	118	**95.4**
s15850	93.6	98.0	2,611,414	2,090,048	53	93.2	1,201,226	**961,792**	77	**96.9**
s38417	98.3	99.3	21,482,240	17,557,920	78	84.9	11,198,720	**9,153,792**	135	**92.1**
s38584	98.9	99.1	17,099,520	12,988,160	66	87.3	9,088,512	**6,903,296**	108	**93.3**

TABLE 4. RESULTS OF ENERGY AND AREA OVERHEAD REDUCTION FOR PREVIOUS WORKS

Circuits	General LFSR	[2]			[7]			[10]		
	ETM	ETM	Energy Reduction (%)	Area Overhead (%)	ETM	Energy Reduction (%)	Area Overhead (%)	ETM	Energy Reduction (%)	Area Overhead (%)
s5378	6,205,783	2,842,162	54.2	9.5	817,907	86.8	14. 1	327,162	94.7	28.3
s9234	6,151,931	3,612,034	41.3	7.7	1,618,041	73.7	11.4	647,216	89.5	37.6
s13207	10,489,489	9,918,453	5.4	4.0	5,735,179	45.3	6.1	2,294,072	78.1	21.8
s15850	9,153,358	8,899,588	2.8	3.7	3,640,491	60.2	5.7	1,456,196	84.1	16.1
s38417	58,217,693	24,609,502	57.7	1.4	17,080,393	70.7	3.2	6,832,157	88.3	8.8
s38584	51,214,830	21,487,640	58.0	1.4	12,592,241	75.4	2.6	5,036,896	90.2	8.4

TABLE 5. RESULTS OF ENERGY AND AREA OVERHEAD REDUCTION FOR THE PROPOSED METHOD

Circuits	General LFSR	Proposed			
	ETM	ETM	Energy Reduction (%)	LFSR bits	Area Overhead (%)
s5378	6,205,783	**175,286**	**97.2**	11	16.6
s9234	6,151,931	**254,084**	**95.9**	11	21.5
s13207	10,489,489	**1,553,330**	**85.2**	12	11.3
s15850	9,153,358	**928,016**	**89.9**	11	8.3
s38417	58,217,693	**10,910,489**	**81.3**	12	3.8
s38584	51,214,830	**4,843,265**	**90.5**	13	4.2

approaches. Since the skipping logic affects the energy consumption of scan-in mode as well as that of scan-out mode, the energy reduction rate was much larger than previous methods.

V. Summary and Conclusions

To minimize the energy consumption during test application for BISTed circuits, we developed a new pattern generator based on a statistical compression code with the modified input reduction scheme and a new test sequence skipping technique.

The proposed approach proved to be an attractive and an effective solution of BIST for low energy tests. Our experimental results demonstrated that the proposed technique significantly reduced the energy consumption and test data volume while at the same time increasing the stuck-at fault coverage. Experimental results gathered on the benchmark circuits have shown the weighted switching activity reduction ranging from 81.3% to 97.2% with increasing stuck-at fault coverage. In addition, the results show that the proposed method achieve higher test data compression with smaller area overhead than previous works.

References

[1] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," Proceedings of IEEE VLSI Test Symposium, pp. 4-9, 1993.

[2] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A Test Vector Inhibiting Technique for Low Energy BIST Design," Proceedings of IEEE VLSI Test Symposium, pp.407-412, 1999.

[3] F. Corno, M. Rebaudengo and M. S. Reorda, "Low Power BIST via Nonlinear Hybrid Cellular Automata," Proceedings IEEE VLSI Test Symposium, pp. 29-34, 2000

[4] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," Proceedings of IEEE VLSI Test Symposium, pp.4-9, 1993.

[5] H. Cheung and S. Gupta, "A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation," Proceedings of IEEE International Test Conference, pp.386-395, 1996.

[6] R.M Chou, K.K. Saluja, and V.D. Agrawal, "Power Constraint Scheduling of Tests," Proceedings of IEEE International Test Conference, pp.271-274, 1994.

[7] S. Manich, A. Gabarro, M. Lopez and J. Figueras, "Low Power BIST by Filtering Non-Detecting Vectors," Proceedings of IEEE European Test Workshop, pp.165-170, 1999.

[8] B. B. Bhattacharya, S. C. Seth and S. Zhang, "Low Energy BIST Design for Scan Based Logic Circuits," Proceedings of IEEE Conference on VLSI Design, pp. 546-551, 2003.

[9] N. C. Lai, S. J. Wang and Y. H. Fu, "Low Power BIST with Smoother and Scan-Chain Reorder," Proceedings of Asian Test Symposium, pp. , 2004.

[10] S. Zhang, S. C. Seth and B. B. Bhattacharya, "On Finding Consecutive Test Vectors in a Random Sequence for Energy Aware BIST Design," Proceedings of IEEE International Conference on VLSI Design, pp. 491-496, 2005.

[11] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison Wesley Publishing Company, second edition, 1994.

[12] A. Chandrakasan, T. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," Journal of solid State Circuits, Vol. 27, No. 4, pp.473-484, 1992.

[13] S. Sankaralingarm, R. R. Oruganti and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," Proceedings of IEEE VLSI Test Symposium, pp. 35-40, 2000.

[14] C. A. Chen and S. K. Gupta, "Efficient BIST TPG Design and Test Set Compaction via Input Reduction," IEEE Transactions on Computer Aided Design of Integrated Circuit and Systems, Vol. 17, pp.692-705, 1998.

[15] S. Chun, Y. Kim, J. Im and S. Kang, "MICRO: A New Hybrid Test Data Compression/Decompression Scheme," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 6, pp. 649-654, 2006.

[16] S. Kajihara, K. Miyase, "On Identifying Don't Care inputs of Test Patterns for Combinational Circuits," Proceedings of IEEE International Conference on Computer Aided Design, pp.364-369, 2001.

[17] K. J. Balakrishnan and N. A. Touba, "Relationship Between Entropy and Test Data Compression," IEEE Transactions on Computer Aided Design of Integrated Circuit and Systems, Vol. 26, pp. 386-395, 2007.

[18] H. J. Wunderlich, "Self Test Using Unequiprobable Random Patterns," Proceedings of IEEE International Symposium on Fault Tolerant Computing, pp.258-263, 1987.

[19] E. Kalligeros, X. Kavousianos, D. Bakalis, and D. Nikolos, "New reseeding technique for LFSR-based test pattern generation," Proceedings of IEEE On-line Testing Workshop, pp.80-86, 2001.

[20] Mentor Graphics, "LBISTArchitect Reference Manual," Version 8.9_1