IEICE TRANS. INF. & SYST., VOL.E93–D, NO.1 JANUARY 2010

PAPER Special Section on Test, Diagnosis and Verification of SOCs

Scan Chain Ordering to Reduce Test Data for BIST-Aided Scan Test Using Compatible Scan Flip-Flops

Hiroyuki YOTSUYANAGI[†a)], ***Member***, **Masayuki YAMAMOTO**[†*], ***Student Member***, ***and*** **Masaki HASHIZUME**[†], ***Member***

SUMMARY In this paper, the scan chain ordering method for BIST-aided scan test for reducing test data and test application time is proposed. In this work, we utilize the simple LFSR without a phase shifter as PRPG and configure scan chains using the compatible set of flip-flops with considering the correlations among flip-flops in an LFSR. The method can reduce the number of inverter codes required for inverting the bits in PRPG patterns that conflict with ATPG patterns. The experimental results for some benchmark circuits are shown to present the feasibility of our test method.

key words: *BIST-aided scan test, scan chain ordering, test data reduction, compatible flip-flops, test pattern generation*

1. Introduction

For testing highly integrated circuits, scan design is widely used to achieve high fault coverage for sequential circuits. Since much test data is required for scan design to set internal states of the circuit, reducing test cost such as test data volume and test application time is required. One of the techniques to reduce the test data volume is built-in self-test (BIST) that has test pattern generator such as linear feedback shift register (LFSR) inside the chip. For testing sequential logic circuit, scan-based BIST methods have been proposed. Since there exist some faults called random pattern resistant faults for which only few input patterns exist that can detect these faults, random pattern generated by LFSR can not achieve high fault coverage in practical time. Hence several enhancements of BIST design have been proposed. Some of the methods control the clock for the LFSR to hold some input vectors for reducing test data volume and scan power [1], [2] while the conventional methods provide all of test clocks to LFSR. A BIST-aided scan test (BAST) architecture [3], [4] is one of the techniques that has an inverter block to flip a random pattern to a deterministic ATPG pattern. The method proposed in [2] reduces the control data for the inverter block in BAST by holding the internal state of the inverter block for some test vectors. In this work, to reduce the test data in the BAST architecture, compatible sets of flip-flops are utilized to reduce the data required to flip the bit pattern of the random pattern. A compatible set of flip-flops is defined as a set of flip-flops for which same values can be assigned in any test vectors. It is utilized for configuring multiple scan chain like Illinois scan and scan tree architectures [5]–[7]. In this work, arrangements of flip-flops in BAST are determined based on the compatible sets of flip-flops. Since the flip-flops are placed considering the correlation between the patterns of flip-flops in an LFSR, PRPG can be constructed without a phase shifter in our method.

In Sect. 2, a basic concept of BAST and some test data volume reduction methods on the BAST architecture are described. Section 3 shows our method that utilizes compatible sets of flip-flops to arrange scan flip-flops on scan chains for the BAST architecture. Section 4 shows the procedure of our method. The experimental results for benchmark circuits are shown in Sect. 5 and Sect. 6 concludes the paper.

2. Preliminaries

2.1 BIST-Aided Scan Test

To apply deterministic test pattern using BIST, BIST-aided scan test (BAST) architecture has been proposed in [3], [4]. Figure 1 shows the BIST-aided scan test architecture. The BAST architecture consists of pseudo random pattern generator (PRPG), inverter block, X-masking block, multiple input signature register (MISR) and decoder block. In the BAST architecture, deterministic ATPG patterns are provided using PRPG and the inverter block. The bits in PRPG

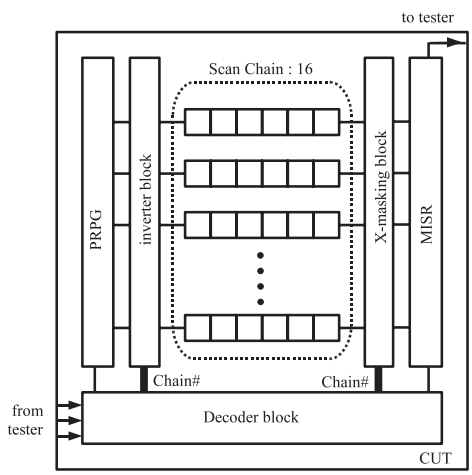

Fig. 1 BIST-aided scan test.

Manuscript received February 6, 2009.
Manuscript revised May 29, 2009.

[†]The authors are with the Dept. of Information Solution, Institute of Technology and Science, University of Tokushima, Tokushima-shi, 770–8506 Japan.
[*]Presently, with the Panasonic Electric Works Co., Ltd.
a) E-mail: yanagi4@ee.tokushima-u.ac.jp

DOI: 10.1587/transinf.E93.D.10

patterns that conflict with the ATPG patterns are flipped using the inverter block. The control signal, that is called BAST code, from tester is provided to the decoder block and decides which bit to be flipped. Since there exist many don't care bits in test pattern in general, the approach can reduce much test time and test data volume with high fault coverage. In this study the reduction of the test data for the input pattern are targeted.

For each scan chain, the ATPG patterns are generated from random patterns by flipping the conflicted bits using the inverter block. Figure 2 shows an example of an inverter block in BAST. A PRPG pattern is provided to scan chains through the inverter block. In this example, an ATPG pattern (1, 0, 0, 1) is provided to the scan chains by flipping the second and the third bits.

We define inverter code as the internal state of the flip-flops in the inverter block that determines which bits of a PRPG pattern to be flipped. Figure 3 shows an input part of the BAST architecture used in our method. In the BAST circuit, a sixteen-bit LFSR and sixteen scan chains are used. The group of the flip-flops at the same position on the scan chains is called a scan slice as shown in the figure. For each scan slice, BAST codes are applied to the decoder block to determine which bits in an inverter code are set to be flipped. The bit of a PRPG pattern is inverted and provided to the scan slice if the corresponding bit in the inverter code has value 1. In conventional BAST test method, all bits in an inverter code are reset for each scan slice.

In an example shown in Fig. 3, the bits of the PRPG pattern at scan chain 3 in scan slice 2 and at scan chain 1 in scan slice 3 are required to be inverted for providing the ATPG pattern to the scan chains. When an input pattern is applied to scan slice 2, first, the BAST code is applied to set inverter code (0, 0, 1, . . . , 0) to the inverter block. The BAST code indicates the address of the scan chain to be flipped. The inverter block flips the bit on the scan chain 3 of the PRPG pattern (0, 1, 1, . . . , 0) for the scan slice 2, therefore ATPG pattern (0, 1, 0, . . . , 0) can be provided to the circuit. The PRPG can not achieve high fault coverage if there exist many random pattern resistant faults that have few input vectors to detect. The inverter block can produce the test vectors for such faults in short test application time.

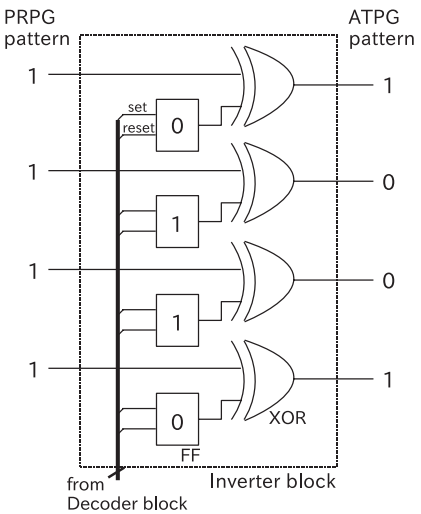

Fig. 2 Inverter block.

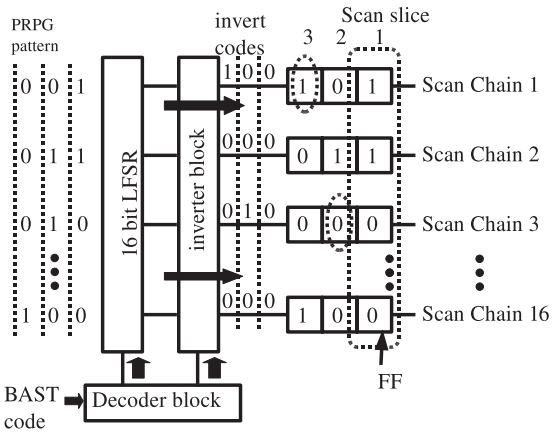

Fig. 3 Input part of BIST-aided scan test.

2.2 Reducing Test Data Volume in a BAST Architecture

The conventional method of the BAST architecture, an inverter code in the inverter block is reset for each scan slice. The flipping bits are decided by BAST codes provided from a tester. One BAST code is required for each one bit to be flipped. In general, ATPG patterns contain many don't care values, hence there are not so many bits to be flipped. However, some PRPG patterns exist such that many bits are flipped due to random pattern resistant faults.

In [2], to reduce the test power in scan-based BIST, the method that keeps an inverter code for some scan slices has been proposed. The method can also be utilized for reducing the BAST codes. To reduce the number of BAST codes, an inverter code does not reset if some bits required to be flipped through some consecutive scan slices in a test pattern. The flip-flops in the inverter block are replaced with toggle-type flip-flops to keep inverter codes until toggle signal is provided from the decoder block.

Figures 4 and 5 show an example of the method to keep an inverter code. Figure 4 shows a PRPG pattern generated by an LFSR and an ATPG pattern selected to provide to the scan chains. The ATPG pattern contains don't care values (denoted by X). Figure 5 shows the inverter code that is required to generate the ATPG pattern from the PRPG pattern in Fig. 4. In this figure, the number of BAST codes required when resetting the inverter codes is shown at the row *reset*. Similarly, the number of BAST codes required when keeping the inverter codes is shown at the row *keep*. For scan slice 1, the assigned ATPG pattern is (X, 0, 0, X) while the PRPG pattern is (1, 1, 1, 1). There exist two care bits in the ATPG pattern. Since all care bits in the ATPG pattern conflict with the PRPG pattern, the inverter code is (0, 1, 1, 0) as shown in Fig. 5. The internal state of the inverter block is reset at the beginning, hence two BAST codes are required

3	2	1	State
1	0	1	LFSR_FF 1
0	1	1	LFSR_FF 2
1	0	1	LFSR_FF 3
0	0	1	LFSR_FF 4

(a) PRPG pattern

3	2	1	Scan slice
X	X	X	Scan chain 1
0	0	0	Scan chain 2
X	0	0	Scan chain 3
1	0	X	Scan chain 4

(b) ATPG pattern

Fig. 4 An example of test patterns for BAST.

	3	2	1	Scan slice
inverter codes	0	0	0	Scan chain 1
	0	1	1	Scan chain 2
	0	1	1	Scan chain 3
	1	0	0	Scan chain 4
the number of BAST codes	1	2	2	reset
	3	0	–	keep

Fig. 5 The inverter code for patterns in Fig. 4.

to set the inverter code for scan slice 1. This case is the same as the example previously shown in Fig. 2 in Sect. 2.1.

For scan slice 2, there exist two conflict bits between the ATPG pattern (X, 0, 1, 0) and the PRPG pattern (0, 1, 0, 0). The inverter code for the scan slice is (0, 1, 1, 0). In conventional method, since the inverter code is reset for every scan slice, two BAST codes are required to set the inverter code the same as for scan slice 1. However, if the internal state of the inverter code is allowed to be held, the number of required BAST codes is 0 because the inverter code for scan slice 2 is the same as that for scan slice 1.

For scan slice 3, only one bit of PRPG pattern conflicts with the ATPG pattern. The inverter code for the scan slice is (0, 0, 0, 1). Whether to keep an inverter code or not is decided by calculating the number of BAST codes required for each cases. In this example, if the inverter code is kept, the difference from the previous inverter code is three bits hence the number of BAST codes is three. For the case that the inverter block is reset, the number of BAST codes required to set the inverter code is one. Therefore, reset is applied and then one BAST code is applied to set the inverter code for scan slice 3.

As shown in this example, reset is not applied when the number of BAST codes can be reduced by keeping an inverter code.

3. Compatible Scan Flip-Flops and Arrangements of Scan Chain

To reduce the test data volume in scan test, several methods have been proposed such that input pattern is shared with multiple scan chains in [5]–[8]. In these methods, the flip-flops placed in parallel are selected among the compatible

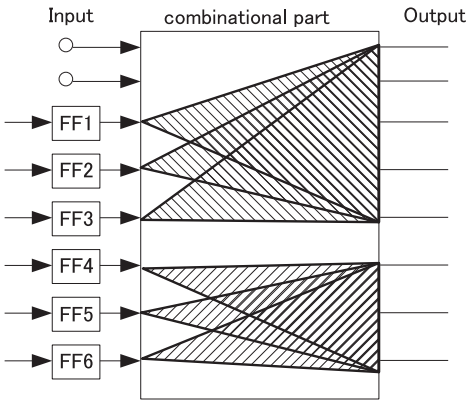

Fig. 6 Calculation of compatible set of flip-flops based on circuit structure.

flip-flops. Two flip-flops FF_a and FF_b are said to be *compatible* for a given test pattern T if they are assigned to the same values or at least one of the flip-flops has don't care for each vector in T [9]. A compatible set of flip-flops is a set of flip-flops in which any pair of flip-flops is compatible. In this work, we calculate compatible sets of scan flip-flops as the set of flip-flops in which the flip-flops have no overlap in the set of the primary outputs and the pseudo outputs that has a path from them by the method in [6]. Figure 6 shows some flip-flops that have overlap in the set of the primary outputs and the pseudo outputs that have a path from them. Flip-flops {FF1, FF2, FF3} are connected to the same set of outputs. Flip-flops {FF4, FF5, FF6} are also connected to the same set of outputs. Since there exists no overlap between the outputs of {FF1, FF2, FF3} and that of {FF4, FF5, FF6}, compatible sets of scan flip-flops can be obtained by selecting one FF from each group. For example, a set of flip-flops $FF1, FF4$ is compatible flip-flops since there exists no outputs that has paths from both FF1 and FF4. This means that no faults require the assignment of the inputs for both FF1 and FF4. Since this method only find compatible sets from the circuit structure, the obtained compatible sets do not depend on test pattern.

If an LFSR is used as PRPG, there exists correlation between the inputs provided from the LFSR to scan chains. In general, to avoid the correlation a phase shifter constructed of XOR gates is inserted between an LFSR and the input of scan chains [10]. In the BAST architecture, the PRPG pattern can be modified through an inverter block even if correlation between the patterns exists. In this work, we utilize the simple LFSR as PRPG and configure scan chains using the compatible set of flip-flops with considering the correlations among flip-flops in an LFSR. Figure 7 shows a random pattern produced by a simple LFSR. In Fig. 7, a value at a bit is tend to be the same as the value at the neighboring bit at the previous clock. Then, we propose the method to arrange scan flip-flops in scan chains based on compatible scan flip-flops as shown in Fig. 8. In Fig. 8, the sets of flip-flops marked as group A, B, C, or D are tend to be set to the same value by LFSR. In the procedure, the flip-flops in a

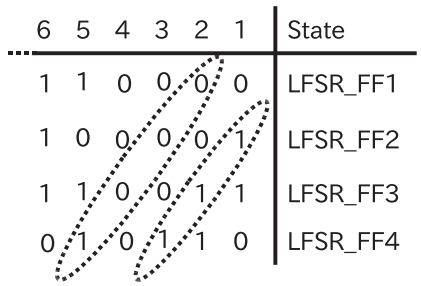

Fig. 7 The random pattern produced by a simple LFSR.

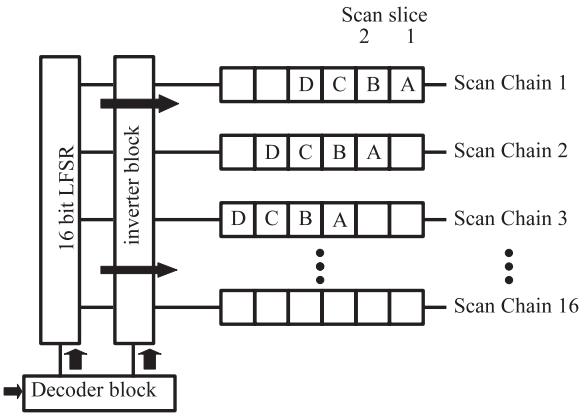

Fig. 8 Compatible FF arrangement in scan chain.

compatible set are arranged diagonally on the scan chains. Even if the LFSR provides the same value at these flip-flops for every test vectors, test patterns can be generated with no conflicts with the LFSR pattern at these flip-flops. If most of the test patterns can be superimposed to the LFSR patterns, the number of the BAST codes can be reduced.

4. Procedure

In our procedure, compatible sets of flip-flops are calculated based on circuit structure using the method in [6]. The BAST architecture is applied to the circuit with the scan chains. The test patterns generated by ATPG are superimposed into the random pattern obtained by an LFSR such that there are less conflicting values between the vectors using a greedy algorithm. Then, the input bits that are required to be flipped at the inverter block are calculated. In the experiments, two kinds of the inverter blocks are utilized for comparison. One is the inverter block that resets inverter code for each test vector. The other is the inverter block that is capable of holding the inverter codes for consecutive scan slices and test vectors. In the case for each scan slice of a test vector whether the reset signal is applied or not is decided by the number of bits required to be flipped.

The procedure for ordering scan chains and generating test pattern for proposed BAST architecture is described as follows.

Procedure: Scan chain ordering and test pattern generation for BAST architecture

1. Compatible sets of flip-flops are calculated such that all flip-flops in a set have no overlap in the set of the outputs connected from the flip-flops using the method in [6].
2. Set $i = 1$, $j = 1$. (i, j) denotes the position of a FF to be arranged. At the next step, the selected FF is placed at scan slice j on scan chain i.
3. The flip-flops in the largest unprocessed compatible set are arranged into the scan chains diagonally. One FF is arbitrarily selected from the compatible set and placed at (i, j). Then both i and j are incremented by one. If $j >$ (number of scan chains), then set $j = 1$. If position (i, j) reaches the end of a scan chain or the position that has already assigned, (i, j) is changed to the unassigned position (i_{min}, j_{min}) on the scan chain j_{min} that has unassigned FFs. In this case, some FFs in the compatible set are not placed diagonally. The other FFs in the compatible set are also arranged similarly.
4. Step 3 is repeated until no unprocessed compatible set remains.
5. Flip-flops that are not involved in the compatible sets are arranged into the remained positions in scan chains.
6. ATPG patterns are generated for the scan chains.
7. The PRPG patterns produced by the LFSR are simulated such that the test length is the same as the ATPG patterns.
8. An ATPG pattern and a PRPG pattern are selected such that the least number of BAST codes are required for inverting the PRPG pattern to apply the ATPG pattern. This step is repeated for all ATPG patterns and determines the order of ATPG patterns superimposed into the PRPG patterns.
9. The inverter codes are calculated for each pair of PRPG and ATPG patterns.
10. For each PRPG pattern, BAST codes are generated for each scan slice based on the inverter codes.

In step 3, the arrangements of the flip-flops in a compatible set can be replaced with each other. The replacement will be applied when the layout constraints such as routing constraints are set for the flip-flops. If some flip-flops do not meet such constraints, these flip-flops are separated from the compatible set. In step 8, our previous method in [11] determined the pairs of ATPG and PRPG patterns by only calculating the number of conflicting bits between the vectors and did not consider the previous inverter codes. In this procedure, since the pairs of ATPG and PRPG patterns are selected based on the number of BAST codes calculated for both with reset and without reset, the better pairs can be selected to obtain less BAST codes. Figure 9 shows an example of a pair of PRPG pattern and ATPG pattern selected in the procedure. The ATPG pattern with least don't care bits is selected first. In this example, the ATPG pattern TA_0 is selected as the first pattern to find the PRPG pattern to be superimposed. The number of required BAST codes is calculated for each PRPG pattern. In this case, the number of BAST codes required for the pairs (TA_0, TP_0), (TA_0, TP_1)

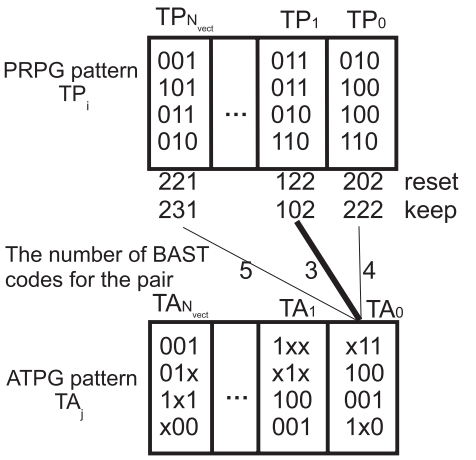

Fig. 9 Selection of a pair of PRPG and ATPG patterns.

Table 1 Test pattern generation for BAST.

circuit	len	vect	inv	test data	test time
s5378	14	110	6,124	45,984	7,788
s9234.1	16	147	7,363	58,290	9,878
s13207.1	44	271	35,813	286,422	48,052
s15850.1	39	145	10,444	96,594	16,283
s35932	111	17	7,200	54,522	9,215
s38417	104	103	50,705	368,502	61,624
s38584.1	92	126	17,893	176,910	29,703
b04s	5	47	557	4,752	844
b07s	4	41	301	2,790	510
b12	8	98	1,098	11,292	1,988
b13s	4	30	232	2,112	386
b14s	18	440	10,480	110,400	18,858
b15s	31	437	8,884	134,586	22,899
b17s	91	445	30,266	424,566	71,297
b20s	33	405	24,752	228,702	38,555
b21s	33	411	23,355	221,508	37,362
b22s	48	439	35,637	340,254	57,196

and $(TA_0, TP_{N_{vect}})$ are 4, 3 and 5, respectively. Please note that though the least number of the inverted bits is obtained for the pair (TA_0, TP_0), the pair (TA_0, TP_1) is selected in this step since the BAST codes can be reduced to 0 for scan slice 2 with keeping the inverter codes.

The test data volume is calculated by the formula similar to [2]: $(N_{vect} \times N_{len} + N_{inv}) \times (2 + \log_2 N_{ch})$, where N_{vect} is the number of test vectors, N_{len} is the length of scan chains, N_{inv} is the number of BAST codes required for setting inverter codes, N_{ch} is the number of scan chains. The term $(2+\log_2 N_{ch})$ denotes the number of bits required for a BAST code. Two bits are required to control the inverter block to reset, keep the invert code or set the bit to be flipped. $(\log_2 N_{ch})$ bits are the address of a scan chain to be set.

The test application time is calculated by $(N_{len} + 1) \times N_{vect} + N_{inv} + N_{len}$. In the BAST architecture, the scan-in operation to provide one test vector for the flip-flops takes N_{len} clocks. And one clock is required for the normal operation for each test vector. N_{inv} BAST codes are required to set the inverter codes for the inverter block. When a test vector is provided to scan chains, output responses captured by the scan flip-flops are scanned out. After all test vectors are applied, the scan-out operation is applied by N_{len} clocks to observe the last output response. Therefore, the sum of test time can be given by $(N_{len} + 1) \times N_{vec} + N_{inv} + N_{len}$.

5. Experimental Results

The proposed procedure was applied to some benchmark circuits. In our experiments 16-bit LFSR is used with 16 scan chains. Table 1 shows the experimental results obtained for conventional BAST. Column *len* shows the length of the scan chains. Column *vect* shows the number of ATPG vectors generated for the circuits. Column *inv* shows the number of the inverted bits that conflicted between the random patterns obtained by the LFSR and ATPG patterns. Column *test data* and *test time* show the test data volume and the test application time, respectively. Since the ATPG patterns utilized by the experiments contain many care bits in the experiments due to our ATPG tool, the test data is not optimized for some circuits. The number of the conflict bits can be reduced much if more don't care bits can be found in the test pattern.

Table 2 shows the results obtained for the case that the inverter block is capable of keeping the inverter codes similar to one of the methods in [2]. In this experiment, for each ATPG pattern the number of BAST codes is estimated both by resetting the inverter codes and by keeping inverter codes. The pair of ATPG pattern and PRPG pattern is selected such that the number of BAST codes is minimized. Hence the pairs of ATPG pattern and PRPG pattern differ from the results in Table 1. The number of inverted bits can be reduced by keeping the invert codes for some test vectors. Column *reset* and Column *keep* show the number of BAST codes with resetting and with keeping the inverter codes, respectively. The percentage shown in Column *test data* and *test time* shows the comparison to the results obtained for the conventional BAST shown in Table 1. It is calculated by the following equation. $(\%) = \frac{(conventional)-(results)}{(conventional)}$. In the experiment, the test data volume can be reduced 7.71%, and the test application time can be reduced 7.51% of that for conventional BAST methods on the average.

Table 3 shows the experimental results obtained for scan chains considering compatible FFs. The percentage shown in Column *test data* and *test time* shows the comparison to the results obtained for the conventional BAST shown in Table 1. For many cases the arrangement of flip-flops using compatible flip-flops can attain the higher reduction rate than that for Tables 1 and 2. Our method can reduce 9.56% of the test data volume, and 9.31% of the test application time required for conventional BAST methods on the average. To compare the results with the case that a phase shifter is used, the simulation was applied for the case that the random patterns produced by the software are applied as PRPG patterns instead of a LFSR. Table 4 shows the results obtained using the method that keeps the inverter codes and the PRPG that is not a simple LFSR such as phase shifter. Our method can achieve much reduction of the BAST codes

Table 2 Test pattern generation for BAST with keeping inverter codes.

circuit	vect	inv	reset	keep	test data	%	test time	%
s5378	110	5,133	953	587	40,038	12.93	6,797	12.72
s9234.1	147	6,242	1,461	891	51,564	11.54	8,757	11.35
s13207.1	271	30,085	7,885	4,039	252,054	12.00	42,324	11.92
s15850.1	145	9,255	4,684	971	89,460	7.39	15,094	7.30
s35932	17	6,286	1,466	421	49,038	10.06	8,301	9.92
s38417	103	43,818	7,109	3,603	327,180	11.21	54,737	11.18
s38584.1	126	15,850	10,135	1,457	164,652	6.93	27,660	6.88
b04s	47	484	186	49	4,314	9.22	771	8.65
b07s	41	267	130	34	2,586	7.31	476	6.67
b12	98	969	661	123	10,518	6.85	1,859	6.49
b13s	30	210	93	27	1,980	6.25	364	5.70
b14s	440	9,517	7,277	643	104,622	5.23	17,895	5.11
b15s	437	7,890	12,827	720	128,622	4.43	21,905	4.34
b17s	445	27,989	38,792	1,703	410,904	3.22	69,020	3.19
b20s	405	22,571	11,852	1,513	215,616	5.72	36,374	5.66
b21s	411	21,444	12,156	1,407	210,042	5.18	35,451	5.11
b22s	439	32,491	18,952	2,120	321,378	5.55	54,050	5.50

Table 3 Test pattern generation for BAST with scan chains considering compatible FFs.

circuit	vect	inv	reset	keep	test data	%	test time	%
s5378	106	4,894	842	642	38,268	16.78	6,498	16.56
s9234.1	139	5,978	1,385	839	49,212	15.57	8,357	15.40
s13207.1	267	29,866	7,954	3,794	249,684	12.83	41,925	12.75
s15850.1	133	8,984	4,344	843	85,026	11.98	14,343	11.91
s35932	16	6,345	1,275	501	48,726	10.63	8,248	10.49
s38417	103	43,798	6,321	4,391	327,060	11.25	54,717	11.21
s38584.1	136	16,454	11,090	1,422	173,796	1.76	29,194	1.71
b04s	49	488	199	46	4,398	7.45	787	6.75
b07s	44	257	147	29	2,598	6.88	481	5.69
b12	98	961	648	136	10,470	7.28	1,851	6.89
b13s	31	211	101	23	2,010	4.83	370	4.15
b14s	426	8,597	6,698	970	97,590	11.60	16,709	11.40
b15s	414	6,982	12,288	546	118,896	11.66	20,261	11.52
b17s	437	26,851	37,376	2,391	399,708	5.85	67,146	5.82
b20s	401	21,246	10,639	2,594	206,874	9.54	34,913	9.45
b21s	413	20,227	11,054	2,575	203,136	8.29	34,302	8.19
b22s	436	31,036	17,441	3,487	311,784	8.37	52,448	8.30

Table 4 Test pattern generation for BAST with keeping inverter codes based on random patterns.

circuit	vect	inv	reset	keep	test data	%	test time	%
s5378	110	5,168	979	561	40,248	12.47	6,832	12.28
s9234.1	147	6,247	1,503	849	51,594	11.49	8,762	11.3
s13207.1	271	30,340	7,685	4,239	253,584	11.46	42,579	11.39
s15850.1	145	9,273	4,686	969	89,568	7.27	15,112	7.19
s35932	17	6,430	1,446	441	49,902	8.47	8,445	8.36
s38417	103	43,895	7,085	3,627	327,642	11.09	54,814	11.05
s38584.1	126	15,927	10,136	1,456	165,114	6.67	27,737	6.62
b04s	47	507	190	45	4,452	6.31	794	5.92
b07s	41	271	135	29	2,610	6.45	480	5.88
b12	98	976	646	138	10,560	6.48	1,866	6.14
b13s	30	226	106	14	2,076	1.7	380	1.55
b14s	440	9,608	7,358	562	105,168	4.74	17,986	4.62
b15s	437	8,189	12,859	688	130,416	3.1	22,204	3.04
b17s	445	28,264	38,684	1,811	412,554	2.83	69,295	2.81
b20s	405	22,597	11,805	1,560	215,772	5.65	36,400	5.59
b21s	411	21,345	12,152	1,411	209,448	5.44	35,352	5.38
b22s	439	32,637	18,940	2,132	322,254	5.29	54,196	5.25

with a simple LFSR. The results obtained for the proposed method can achieve about 2% of reduction in test data and test time compared with the previous method. To generate the test pattern such that the more same values are assigned for the compatible set of flip-flops that reside diagonally in the scan chains may achieve much reduction in the BAST codes. TPG for the BAST with the scan chain ordering is still left as the future work.

6. Conclusion

The test data and test time reduction method for BIST-aided scan test is proposed. To reduce the conflicts between random patterns generated by LFSR and ATPG patterns, the compatible sets of flip-flops are utilized for scan chain ordering. Along with the inverter block that is capable of keeping the inverter codes, our method can reduce the test data and test time. Since the compatible set of flip-flops can be replaced each other in scan chain, the method to find optimized arrangements for LFSR is still left as the future work.

References

[1] Y. Kim, K. Kim, I. Kim, H. Son, and S. Kang, "A new scan power reduction scheme using transition freezing for pseudo-random logic BIST," IEICE Trans. Inf. & Syst., vol.E91-D, no.4, pp.1185–1188, April 2008.

[2] A. Suto, M. Arai, and K. Iwasaki, "Note on test power reduction for scan-based hybrid bist," IEICE Technical Report, DC2007-72, 2008.

[3] T. Hiraide, K.O. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka, and T. Mochiyama, "Bist-aided scan test - A new method for test cost reduction," Proc. VLSI Test Symp., pp.359–364, 2003.

[4] M. Arai, S. Fukumoto, K. Iwasaki, T. Matsuo, T. Hiraide, H. Konishi, M. Emori, and T. Aikyo, "Test data compression for scan-based bist aiming at 100x compression rate," IEICE Trans. Inf. & Syst., vol.E91-D, no.3, pp.726–735, March 2008.

[5] K. Miyase and S. Kajihara, "Optimal scan tree construction with test vector modification for test compression," Proc. Asian Test Symp., pp.136–141, 2003.

[6] H. Yotsuyanagi, T. Kuchii, S. Nishikawa, M. Hashizume, and K. Kinoshita, "On configuring scan trees to reduce scan shifts based on a circuit structure," Proc. DELTA, pp.269–274, 2004.

[7] H. Yotsuyanagi, T. Kuchii, S. Nishikawa, M. Hashizume, and K. Kinoshita, "Reducing scan shifts using configurations of compatible and folding scan trees," J. Electron. Test., Theory Appl., vol.21, no.6, pp.613–620, Dec. 2005.

[8] D. Xiang, S. Gu, J.G. Sun, and Y.L. Wu, "A cost-effective scan architecture for scan testing with non-scan test power and test application cost," Proc. DAC, pp.744–747, June 2003.

[9] I. Hamzaoglu and J.H. Patel, "Reducing test application time for full scan embedded cores," Proc. Int'l Symp. Fault-Torelant Computing, pp.260–267, 1999.

[10] V.D. Agrawal, C.R. Kime, and K.K. Saluja, "A tutorial on built-in self-test, part 2: Applications," IEEE Des. Test Comput., vol.10, no.2, pp.69–77, 1993.

[11] M. Yamamoto, H. Yotsuyanagi, and M. Hashizume, "Scan chain configuration for bist-aided scan test using compatible scan flip-flops," Workshop on RTL and High Level Testing, 2008.

Hiroyuki Yotsuyanagi received his B.E., M.E. and Ph.D degrees from Osaka University, in 1993, 1995 and 1998, respectively. He is currently an Associate Professor of the Department of Electrical and Electronic Engineering, the Univ. of Tokushima. His research interest includes test synthesis for sequential circuits and current testing for CMOS ICs. He is a member of the IEEE.

Masayuki Yamamoto received his B.E. and M.E. degree in electrical engineering from the Univ. of Tokushima, in 2007 and 2009, respectively. He is currently working with the Panasonic Electric Works Co., Ltd. His research interest includes design for testability of sequential circuits.

Masaki Hashizume received his B.E. and M.E. degrees in electrical engineering from the Univ. of Tokushima and Dr.E. degree from Kyoto Univ., in 1978, 1980 and 1993, respectively. He is currently a Professor of the Department of Electrical and Electronic Engineering, Faculty of Engineering, the Univ. of Tokushima. His research interests are logic synthesis and supply current testing of logic circuits.