

Brief Contributions

Reconfigured Scan Forest for Test Application Cost, Test Data Volume, and Test Power Reduction

Dong Xiang, *Senior Member, IEEE*,
Kaiwei Li, Jiaguang Sun, and
Hideo Fujiwara, *Fellow, IEEE*

Abstract—A new scan architecture called reconfigured scan forest is proposed for cost-effective scan testing. Multiple scan flip-flops can be grouped based on structural analysis that avoids new untestable faults due to new reconvergent fanouts. The proposed new scan architecture allows only a few scan flip-flops to be connected to the XOR trees. The size of the XOR trees can be greatly reduced compared with the original scan forest; therefore, area overhead and routing complexity can be greatly reduced. It is shown that test application cost, test data volume, and test power with the proposed scan forest architecture can be greatly reduced compared with the conventional full scan design with a single scan chain and several recent scan testing methods.

Index Terms—Scan forest, test application cost, test data volume, test power.

1 INTRODUCTION

SCAN design makes test generation of a circuit the same as that of a combinational one, which can obtain complete fault coverage and make test application cost, test data volume, and test power prohibitively high [1], [5], [11], [23], [27]. Test application cost, test power, and test data volume can be reduced by efficient test generation schemes [16], [21], [22], test compression techniques [2], [7], [8], scan flip-flop ordering [10], clock disabling [9], [19], extra logic insertion [13], [19], test input reduction [6], and new scan architectures [3], [4], [12], [15], [18], [19], [25], [28].

Test compression techniques are used to reduce test application time, test power, and test data volume. Chandra and Chakrabarty introduced a method to reduce test application cost and test data volume via test data compression based on Golomb encoding scheme [7] and frequency-directed run-length codes [8]. A test pattern compression methodology [2] was presented with a small number of virtual scan chains, which were used to drive a large number of internal scan chains and reduce the test data volume significantly. Wohl et al. [24] designed a noncancelling compactor by connecting each input of the compactor to multiple outputs. They also proposed a compactor to support diagnosis by using extra inputs to control different phases in the testing process. Mitra and Kim [17] presented a test response compactor that tolerates don't cares in test responses. The method was proposed based on a \times -compactor matrix, where all rows of the matrix represent the compactor inputs and columns represent the compactor outputs. Similarly, the \times -compactor supports diagnosis.

A lot of new scan architectures have been proposed to reduce test application cost and test power. Lee et al. [15] presented a

novel test application scheme by driving multiple scan chains with a single scan-in signal in a core-based system, which can reduce test application time drastically without any degradation on test efficiency. The Illinois scan architecture drives multiple scan segments by a single scan-in [12] in the same circuit. This technique can effectively reduce test application time, which may also cause some degradation on fault coverage or test efficiency. Serial scan shifts may be necessary for some test vectors. A new scan architecture called *scan forest* was proposed to reduce test application time, test power, and test data volume recently in [25]. It is found that many scan flip-flops can be assigned the same value for all test vectors. No new reconvergence generates when merging the group of scan flip-flops. The predecessor of the group is a scan flip-flop selected from another group of scan flip-flops. All scan flip-flop groups form multiple scan trees. However, the scan forest connects most of the scan flip-flops to XOR trees that may cause area and routing complexity problems. Another problem is that random construction of the XOR trees in [25] can produce fault aliasing. Various tree-based scan architectures were proposed to reduce test application time and test power in [3], [4], [18], [28] recently.

The main differences between the proposed scan forest architecture and the Illinois scan architecture should be 1) construction of the scan architectures, 2) scan shifting scheme, and 3) the test response compaction scheme. The Illinois scan architecture groups scan flip-flops based on test vector analysis, where it contains one scan-in signal and the same number of scan flip-flops at any level. The scan forest contains multiple scan-in signals, where it is not necessary for each level to have the same number of scan flip-flops. Scan flip-flops in the scan forest are grouped based on structural analysis. Scan forest does not introduce any aliasing faults, where all test vectors are applied in parallel. However, the Illinois scan architecture needs serial scan shifting, which makes the test data volume and test application time reduction ratio of the Illinois scan architecture not as good as those of the scan forest. As for the test response compaction scheme, the Illinois scan presents an MISR-based scheme. The scan forest compacts test response based on structural analysis, which introduces no aliasing. Area overhead of the test compactor is the same as that of the cancellation compactor, where all inputs are connected to the compactor only once.

Let the *test circuit* be the circuit used to generate tests after design for testability. The *average test power* is test power consumption of all test vectors in the test set divided by the number of tests and the *peak test power* is defined as the maximum test power consumption when applying one test vector in the test set, like [2], [7], [8], [9], [15].

2 SCAN FOREST TO REDUCE TEST APPLICATION COST, TEST DATA VOLUME, AND TEST POWER

Scan forest [26] is proposed to reduce test application cost, test data volume, and test power consumption. The scan forest architecture contains a couple of scan trees. The root of each scan tree is driven by a primary input. In each scan tree, a group of scan flip-flops is driven by the scan-in, where only one of the scan flip-flops in the group drives another group of scan flip-flops, and so on. All leaf scan flip-flops in the scan tree are connected to the XOR trees. Outputs of all XOR trees are multiplexed with primary outputs. Fig. 1 presents a test circuit of the circuit with a scan forest, where scan flip-flops in the same group share the same pseudoprimary input. Pseudoprimary outputs of the leaf scan flip-flops are connected to the XOR trees as presented in Fig. 1. Two scan flip-flops can be put into the same

- D. Xiang, K. Li, and J. Sun are with the School of Software, Tsinghua University, Beijing 100084, P.R. China. E-mail: dxiang@tsinghua.edu.cn.
- H. Fujiwara is with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan.

Manuscript received 13 Sept. 2004; revised 1 June 2006; accepted 4 Oct. 2006; published online 1 Feb. 2007.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0298-0904. Digital Object Identifier no. 10.1109/tc.2007.1002.

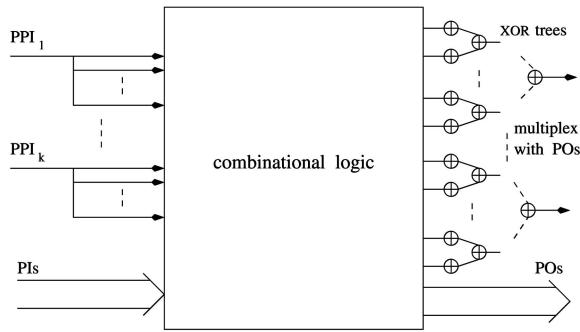


Fig. 1. Test circuit of the reconfigured scan forest designed circuit.

group if they do not have any common successor in the combinational part of the circuit. The original idea of scan flip-flop grouping is from [26], to reduce the number of extra pins.

A table is defined as follows: $conv(i, j) = 1$ if two scan flip-flops i and j have at least one common successor in the combinational part of the circuit; otherwise, $conv(i, j) = 0$. We say two scan flip-flops i and j satisfy the test signal condition if $conv(i, j) = 0$. Another table is defined as follows: $pred(i, j) = 1$ if two scan flip-flops i and j have a common predecessor in the combinational part of the circuit; otherwise, it is equal to 0. Two scan flip-flops i and j are said to meet the test response compaction condition if $pred(i, j) = 0$.

The table $conv(i, j)$ can be obtained as follows: First, get the set of primary outputs or pseudoprimary outputs by traversing all combinational successors of each pseudoprimary input. Consider a scan flip-flop i , each of the remaining scan flip-flops j , set $conv(i, j) = 1$ if i and j reach at least one common primary output or pseudoprimary output. Similarly, the table $pred(i, j)$ can be constructed. The above technique can greatly reduce the CPU time to construct the scan forest architecture [25]. CPU time to construct a reconfigured scan forest is presented in Section 5.

The testing scheme of the fully scanned circuit with scan forest is presented as follows: First of all, set the circuit into test mode and scan in values of the test with respect to the leaf scan flip-flops at the last level into the scan forest, where the number of different values is at most the number of scan trees. After that, the values of the scan flip-flops corresponding to the scan flip-flop groups at the last but one level are scanned into all scan trees. The values of the PPIs of the leaf scan flip-flops are scanned to the next level simultaneously. Continue the above process until all values of the test vector are scanned in. Set the circuit into the functional mode; all flip-flops capture responses from the combinational logic when all values of the PPIs and PIs are applied. After the test responses have been received at primary outputs and scan flip-flops for the first test vector, the next test vector is scanned in as stated above while the test responses captured at the scan flip-flops are shifted out. Continue the above process until all test vectors have been applied. Let $\#vectors$ and $levels$ be the number of test vectors and levels of the scan forest. The test application time TAT for a fully scanned circuit with a scan forest is

$$TAT = \#vectors \cdot (levels + 1) + levels. \quad (1)$$

In most cases, only a very small number of shift cycles is necessary to apply one test into the scan forest because $levels$ can be very small. All scan flip-flops in the same group are assigned the same value for all test vectors; therefore, the test data volume can also be greatly reduced. The scan forest connects all leaf scan flip-flops to the XOR trees, which can cause a routing problem, an area overhead problem, and some aliasing faults.

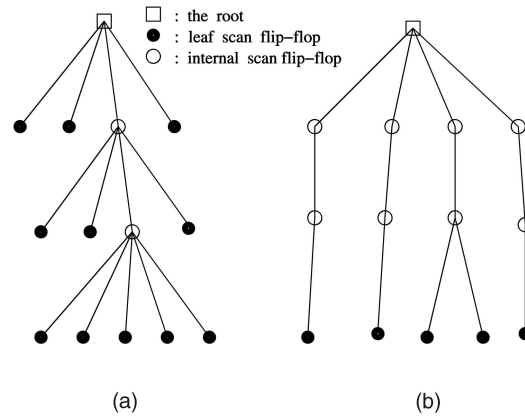


Fig. 2. Reconfigured scan forest: (a) the scan tree and (b) the reconfigured scan tree.

3 XOR TREE CONSTRUCTION TO AVOID FAULT ALIASING

A new scan architecture, called *reconfigured scan forest*, is introduced to resolve the above problems. In the original scan forest [25], outputs of the leaf scan flip-flops are randomly selected to connect to the XOR trees. This scheme can generate a number of aliasing faults. The reconfigured scan forest can greatly reduce the size of the XOR trees, effectively avoiding fault aliasing compared with the scan forest while it reduces test power, test data volume, and test application time like the scan forest [25]. Fig. 2a shows a scan tree with 11 leaf scan flip-flops. The scan tree is reorganized into a new tree with five leaf scan flip-flops, as shown in Fig. 2b. The area overhead of the XOR trees and routing complexity can be greatly reduced. As for two leaf scan flip-flops, f_1 and f_2 , in the same scan tree, there exists a common predecessor v in the scan tree that can only be the root. There exist two paths from both leaf scan flip-flops to the node v in the scan forest $(f_1, v_1, v_2, \dots, v_i, v)$ and $(f_2, v'_1, v'_2, \dots, v'_j, v)$. Generally, it is required that all pairs of flip-flops $(f_1, f_2), (v_1, v'_1), (v_2, v'_2), \dots$, have no common predecessor in the combinational part of the circuit, respectively, in order to avoid fault aliasing.

It is not necessary for each group of flip-flops to meet the above condition when constructing the scan tree properly. As shown in Fig. 3, scan paths 1-3-8 and 2-5-11 are compatible. Let (7, 10) meet

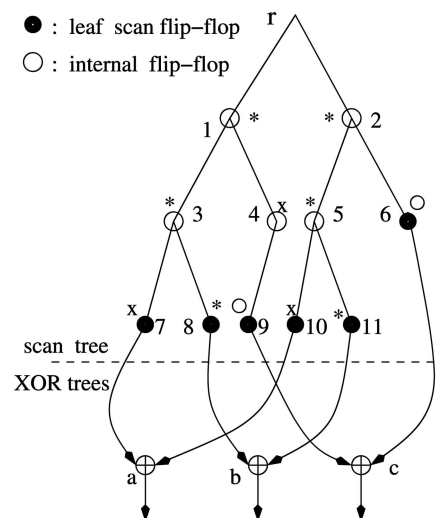


Fig. 3. XOR tree construction in a single scan tree.

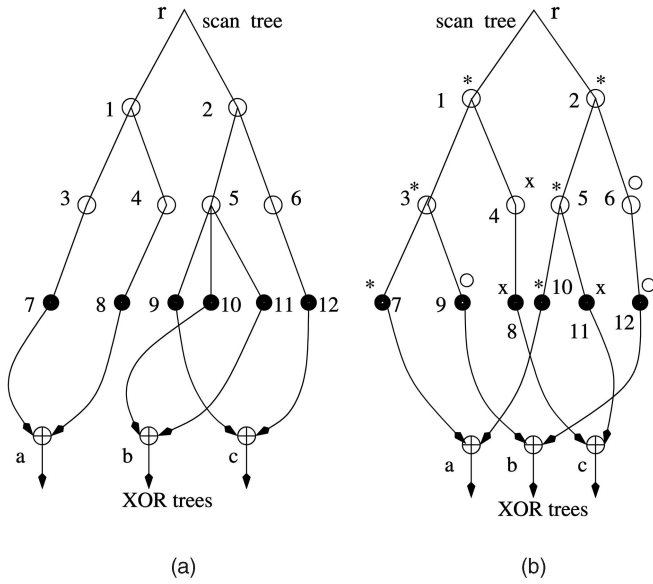


Fig. 4. XOR tree construction to avoid fault aliasing.

the test response compaction condition. It is sufficient for (7, 10) to be connected to the same XOR gate a if 7 and 10 do not have any common combinational predecessor because 7 and 10 are connected with two compatible scan paths. As for XOR gate c , we need to consider flip-flop pairs (6, 9), where the scan flip-flop pair meets the test response compaction condition. However, it is not necessary for flip-flop pair (2, 4) to have no common predecessor because any fault effects captured at 2 can also be observed at the output of gate b . Any fault effects captured at flip-flop 2 can also be propagated through path 2-5-11-b.

There may still exist some aliasing faults even though the above conditions have been met. Let us consider scan flip-flop 1 in Fig. 4. As for any fault effects propagated to scan flip-flop 1, they can also be propagated to scan flip-flops 7 and 8 simultaneously, which are masked by the XOR gate a as presented in Fig. 4a. It is very dangerous to meet this kind of situation because all fault effects propagated to flip-flop 1 are masked. However, this kind of aliasing fault can be avoided by using the following schemes: 1) All leaf scan flip-flops connected to the same XOR tree should be in different scan trees if possible and 2) any pair of leaf scan flip-flops in the same scan tree connected with the same XOR tree should be in different subtrees that are driven by the root directly.

The known compatible scan path information can be used to construct XOR trees. Another useful scheme can be used to construct the XOR trees to avoid fault aliasing. A flip-flop can be exchanged with another unmarked flip-flop at the same level of the scan tree when no common predecessor condition is unable to meet. The above scheme can greatly improve the flexibility to construct the XOR trees. Let scan flip-flop pairs (7, 10), (3, 5), and (1, 2) meet the test response compaction condition as shown in Fig. 4b; connecting scan flip-flops 7 and 10 to the same XOR gate generates no aliasing fault. The scan paths (1, 3, 7) and (2, 5, 10) are compatible. It is better to reconstruct the XOR tree and reconfigured scan tree as shown in Fig. 4b, where scan flip-flop 9 is connected with flip-flop 3 instead of 5. As shown in Fig. 4b, it is necessary for the flip-flop pair (8, 11) to have no common combinational predecessor in order to connect scan flip-flops 8 and 11 with the same XOR gate. As for the XOR tree b , it is enough if scan flip-flops 9 and 12 have no common combinational predecessor without checking scan flip-flop pair (3, 6).

More attention is paid to XOR tree construction for multiple scan trees. As shown in Fig. 5, three leaf scan flip-flops are

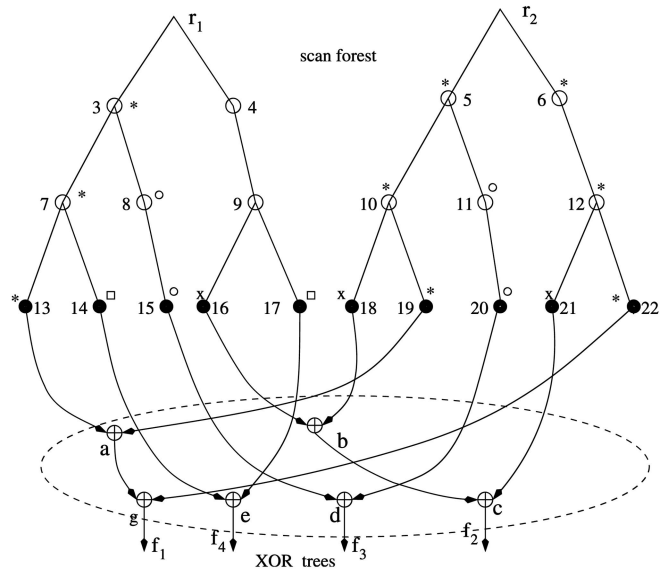


Fig. 5. XOR tree construction in multiple scan trees.

connected to the same XOR tree f_1 . Each pair of scan flop-flops in the groups (13, 19, 22), (7, 10, 12), and (3, 5, 6) do not have any common predecessor in the combinational part of the circuit. Leaf scan flip-flops 14 and 17 can be connected to the same XOR tree if 14 and 17 do not have any common combinational predecessor. Three leaf scan flip-flops, 16, 18, and 21, can be connected to the same XOR tree c only if they do not have any common combinational predecessor. Scan flip-flop pairs (15, 20) and (8, 11) must have no common combinational predecessor in order to construct the XOR tree f_3 .

4 CONCURRENT CONSTRUCTION OF THE RECONFIGURED SCAN FOREST AND XOR TREES

A number of empirical techniques were presented to establish the scan forest and the XOR trees in Section 3. A comprehensive procedure is proposed to construct the scan forest and XOR trees concurrently, as presented in Fig. 6. The procedure calculates the compatible tables $pred()$ and $conv()$ for all scan flip-flop pairs to construct the reconfigured scan forest and the XOR trees based on the schemes introduced in Section 3. The proposed method finds the expected depth of the scan forest and the average number of scan flip-flops in each group according to the greedy scan flip-flop grouping scheme. The greedy scheme puts all scan flip-flops j into the group of scan flip-flop i if $conv(i, j) = 0$ deletes all scan flip-flops selected from the scan flip-flop set. The same scheme is used to group the remaining scan flip-flops until all scan flip-flops have proceeded. A similar scheme is introduced to calculate the table $pred(i, j)$. First, our method gets the set of primary inputs and pseudoprimary inputs that reach all scan flip-flops, using which the table $pred(i, j)$ is calculated. For each pair of scan segments $(i, v_1, v_2, \dots, v_d)$ and $(j, f_1, f_2, \dots, f_d)$ driven by the same scan-in signal, the scan flip-flop pairs $(v_1, f_1), (v_2, f_2), \dots, (v_d, f_d)$ must satisfy the test signal condition. That is, we must have $conv(v_1, f_1) = 0, \dots, conv(v_d, f_d) = 0$. Compared with the greedy scan flip-flop scheme in [25], CPU time to construct the reconfigured scan forest can be greatly reduced.

Two scan segments, $(i, v_1, v_2, \dots, v_d)$ and $(j, v'_1, v'_2, \dots, v'_d)$, can be connected to the same XOR tree if they are compatible, where i and j are different roots of two scan trees, v_1, v_2, \dots, v_d and v'_1, v'_2, \dots, v'_d are scan flip-flops, and v_d and v'_d are leaf scan flip-flops. We call two scan segments $(i, v_1, v_2, \dots, v_d)$ and $(j, v'_1, v'_2, \dots, v'_d)$ compatible if each pair of scan flip-flops

```

reconfigured-scan-forest-construction ( )
{
1. For each pseudo-primary input i, get all primary outputs and pseudo
   -primary outputs that are reachable from i; calculate the table
   conv(i,j) for any pair of scan flip-flops i and j.
2. For each scan flip-flop i, get all primary inputs and pseudo-primary
   inputs that reach i; calculate the table pred(i,j) for any pair of scan
   flip-flops i and j.
3. Set the number of scan trees by the greedy scheme based on the
   derived table conv(), and the depth d of the reconfigured scan forest.
4. For each pair of scan segments (v1, ..., vd) and (v'1, ..., v'd), put
   two scan segments into different scan trees and connect the leaf scan
   flip-flops to the same XOR tree if pred(v1, v'1) = 0, ...,
   pred(vd, v'd) = 0.
5. Connect two scan segments to (v1, ..., vd) and (f1, ..., fd) to the
   same scan tree if conv(v1, f1) = 0, ..., conv(vd, fd) = 0.
6. For all remaining scan flip-flops, construct scan segment (v1, ...,
   vk) (d > k) connect it to a scan tree as presented in Fig. 8 if the scan
   flip-flops satisfy the test signal conditions with the last k scan flip-
   flop test signal conditions with the last k scan flip-flop groups in
   the scan tree. Connect the leaf scan flip-flop of the segment with
   an XOR tree if the test response compaction condition is satisfied
   corresponding to the last k scan flip-flops of all scan segments
   connected to the scan tree.
7. Continue the above process until all scan flip-flops have been
   connected to the scan forest.
}
    
```

Fig. 6. Procedure to construct the reconfigured scan forest.

$(v_1, v'_1), (v_2, v'_2), \dots, (v_d, v'_d)$ meets the test response compaction condition. In this case, it is sufficient for the scan flip-flop pairs to meet only the test response compaction condition. Two segments driven by the same root can also be compatible. However, the test signal condition must still be satisfied. As for two scan segments $(i, v_1, v_2, \dots, v_d)$ and $(i, v'_1, v'_2, \dots, v'_d)$, they are compatible if scan flip-flop pairs $(v_1, v'_1), (v_2, v'_2), \dots, (v_d, v'_d)$ meet the test signal condition and the test response compaction condition simultaneously.

Our method first tries to get as many compatible scan segments as possible. It is better to construct compatible scan segments in different scan trees. As shown in Fig. 7, the scan forest contains three scan trees. Our method gets five complete scan segments, two for the first two scan trees each and one for the third scan tree. Two segments $(r_1, 1, 2, 3, 4)$ and $(r_2, 11, 12, 13, 14)$ are compatible

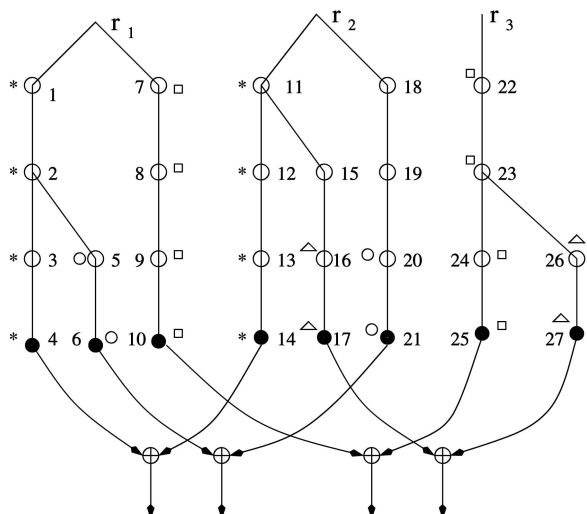


Fig. 7. Example of the reconfigured scan forest.

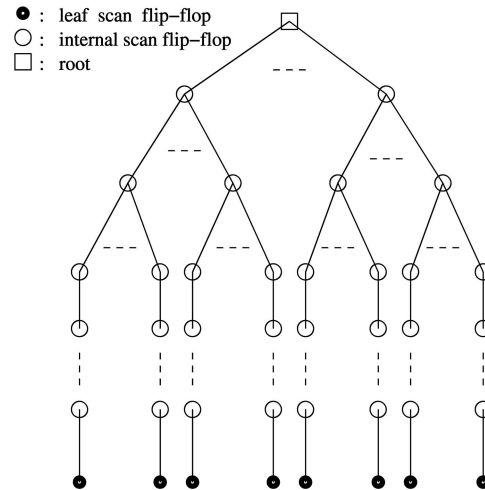


Fig. 8. Scan tree construction for very large circuits.

because scan flip-flop pairs (1, 11), (2, 12), (3, 13), and (4, 14) meet the test response compaction condition. Similarly, compatible scan segment pairs $(r_1, 7, 8, 9, 10)$ and $(r_3, 22, 23, 24, 25)$, and $(r_1, 7, 8, 9, 10)$ and $(r_3, 22, 23, 24, 25)$ are also constructed.

It is not necessary for two scan segments to have the same length to connect them to the same XOR tree. As shown in Fig. 7, two scan segments (5, 6) and (18, 19, 20, 21) are compatible only if (5, 20) and (6, 21) meet the test response compaction condition because scan flip-flop 5 is driven by scan flip-flop 2. It is not necessary for (2, 19) or (1, 18) to meet the test response compaction condition because fault effects of the common predecessors of 2 and 19 or 1 and 18 can be propagated along the scan paths $(r_1, 1, 2, 3, 4)$ to the outputs. As for scan paths $(r_2, 11, 15, 16, 17)$ and $(r_2, 22, 23, 26, 27)$, it is not necessary for the scan flip-flop pairs (23, 15) and (22, 11) to meet the test response compaction condition because the fault effects of their common predecessors can be propagated along the scan path $(r_3, 22, 23, 24, 25)$. All scan flip-flops in the same scan tree and at the same level should meet the test signal condition.

The average size of the scan flip-flop groups at each level of the scan trees is the most important factor of the test data compression ratio. As for the test response compaction ratio, it is determined by $l \cdot n_{xor}$, where l is the depth of the scan trees and n_{xor} is the number of XOR trees plus the number of primary outputs. The average size of the scan flip-flop groups in the scan trees and the number of XOR trees is determined by the structural features of the circuit. No aliasing fault is produced by the reconfigured scan forest and the XOR trees.

When the circuit has a very large number of scan flip-flops (for example, the circuit contains more than 100,000 scan flip-flops), each scan flip-flop group may contain up to 1,000 scan flip-flops. It is not good for each scan-in signal to drive so many scan flip-flops directly. A new scan tree architecture, shown in Fig. 8, can be used. Each node in the scan tree drives a scan chain similar to the reconfigured scan forest after the number of nodes at the same level has been large enough. Therefore, any node in the scan tree does not drive many nodes. This can completely reduce the routing complexity.

5 EXPERIMENTAL RESULTS

The proposed reconfigured scan architecture has been implemented and the ATALANTA test generator is adopted to generate tests [14]. The test circuit without the XOR trees as shown in Fig. 1 is used to generate tests. After the XOR trees have been added to the

TABLE 1
Construction of the Reconfigured Scan Forest for the Benchmark Circuits

circuits	#PO	#FFs	greedy scheme							reconfigured scan forest									
			<i>nef</i>	<i>alias</i>	<i>level</i>	<i>group</i>	<i>size</i>	<i>leaf</i>	AO(%)	<i>S</i> _{in}	<i>group</i>	<i>size</i>	<i>leaf</i>	AO(%)	TA(%)	PTP(%)	TP(%)	CPU(s)	
s1423	5	74	0	23	4	52	7	39	10.56	13	30	2	20	6.72	9.90	6.72	15.58	0.02	
s5378	49	179	5	9	2	61	26	153	9.95	11	66	3	32	4.89	3.67	8.64	8.89	0.03	
s9234	22	228	16	113	5	85	26	162	7.04	15	90	3	41	3.35	3.11	7.45	7.34	0.08	
s13207	121	669	40	53	7	192	139	508	11.40	30	186	4	112	4.17	1.17	1.25	1.19	0.63	
s15850	87	597	35	60	13	182	84	429	9.18	12	101	6	87	4.35	2.30	5.39	2.30	0.6	
s35932	320	1728	868	0	1	11	284	1728	16.17	2	11	158	302	6.33	0.18	0.41	0.32	8.65	
s38417	106	1636	99	1376	4	100	286	1564	14.19	20	84	20	546	5.59	0.23	0.61	0.22	5.0	
s38584	278	1452	6	619	13	146	178	1318	10.94	10	96	16	121	4.05	0.96	1.75	0.98	2.48	
s1269	10	37	0	16	2	29	5	26	7.02	10	30	2	13	5.04	9.90	15.63	15.58	0.0	
s1512	21	57	0	6	2	31	12	55	10.09	8	32	2	16	5.52	7.75	21.06	4.10	0.017	
s3271	14	116	7	23	1	23	25	116	13.13	8	24	5	40	6.03	3.05	10.02	0.98	0.017	
s3330	73	132	0	0	3	84	19	88	4.62	14	84	2	63	4.64	2.99	8.87	3.02	0.033	
s3384	26	183	0	1	2	48	24	178	14.57	8	48	4	33	5.63	3.98	6.11	6.01	0.0	
s4863	16	104	3	1	1	16	51	104	7.54	4	16	7	28	3.24	5.16	13.76	1.38	0.0	
s6669	55	239	0	0	1	48	48	239	10.68	8	48	5	49	4.52	2.57	7.21	6.69	0.07	

TABLE 2
Performance Comparison with the Previous Methods

circuits	reconfigured scan forest			BO [2]		MCD [8]		Golomb [6]		FDR[7]
	TA(%)	TP(%)	TDR(%)	TA(%)	TDR(%)	TA(%)	TP(%)	TP(%)	TDR(%)	TDR(%)
s5378	1.64	1.46	33.68	—	—	21.0	17.5	21.98	36.27	38.68
s9234	3.11	7.34	37.48	—	—	29.4	25.5	33.70	40.15	39.37
s13207	0.99	0.45	28.52	16.20	13.77	6.1	4.2	6.32	15.67	12.33
s15850	1.79	2.84	20.87	21.54	18.29	20.2	15.1	14.73	32.45	28.05
s35932	0.18	0.32	1.14	17.60	16.34	9.2	3.7	7.89	—	—
s38417	0.20	0.31	4.88	20.51	19.07	38.3	22.9	18.65	41.92	34.65
s38584	0.65	1.16	6.58	16.02	14.48	21.0	16.0	16.48	40.39	35.33

TABLE 3
Performance Evaluation of the Largest ITC99 Circuits

circuits	#FFs	full scan				reconfigured scan forest													
		FC	TE	vec	# faults	FC	TE	vec	TDR	TA	TP	groups	AO(%)	<i>S</i> _{in}	size	<i>abt</i>	<i>red</i>	CPU (s)	
b17	1415	99.07	99.2	2255	100756	98.95	99.1	2930	23.83	0.71	1.53	309	3.42	37	5	-29	0	11.8	
b18	3320	99.13	99.15	4697	304964	98.83	98.99	4148	10.71	0.29	0.68	370	3.31	37	10	44	32	127	
b19	6642	99.0	99.47	20312	1355680	98.86	99.34	13722	2.78	0.26	0.75	250	2.21	10	28	-791	32	762	

circuit, the HOPE fault simulator [14] is utilized to do fault simulation using the test vectors on the reconfigured scan forest designed circuit. Table 1 presents results of the ISCAS circuits. As shown in Tables 1 and 2, TA, TP, PTP, *alias*, AO, *nef*, TDR, *leaf*, *group*, and *size* represent test application cost reduction ratio, test power reduction ratio and peak power reduction ratio corresponding to full scan design with a single scan chain, the number of aliasing faults, area overhead, the number of redundant faults in the DFT logic, test data volume reduction ratio, the number of leaf scan flip-flops, the number of scan flip-flop groups, and the maximum size of the scan flip-flop groups, respectively. Parameters *alias* and *nef* are controlled to be 0 by the reconfigured scan forest architecture for all benchmark circuits. It is shown that the number of leaf scan flip-flops greatly decreases after using the reconfigured scan forest architecture compared with the original scan forest. This can also greatly reduce the area overhead and

routing complexity. The CPU time in Table 1 presents the CPU time to group scan flip-flops for the reconfigured scan forest.

As shown in Table 2, the proposed method is compared with a number of recent scan testing methods: the test compression techniques using Golomb codes [7] (Golomb) and the frequency-directed run-length codes [8] (FDR) and the MCD method in [9] on test application cost, test power, and test data volume reduction ratios. The proposed scan architecture gets better TDR than [7], [8] for all circuits except s13207 and much better TDR than [7], [8], [2] for s35932, s38417, and s38584 and better TDR than [2] for all circuits except s13207 and s15850. Compared with MCD [9], the new method gets better TA and TP for all circuits. Compared with [2], [9], our method obtains much better TA for all circuits and much better TP for all circuits than [9], [7].

Table 3 presents results of the reconfigured scan forest on the three largest ITC99 benchmark circuits. All three circuits contain

more than 100,000 single stuck-at faults and b19 has more than 1,300,000 faults. The TetraMax test generation system of SYNOPSIS is used to generate tests for the circuits. All three circuits still have some aborted faults for both the single chain designed and the reconfigured scan forest designed circuits. Parameters *abt* and *red* present an addition of the number of aborted faults and redundant faults with the reconfigured designed circuit compared to that of the single scan chain designed circuit. Two negative numbers for *abt* in Table 3 represent that the number of aborted faults with the reconfigured scan forest decreases corresponding to that of the single scan chain designed circuit. Fault coverage and test efficiency (TE, including the redundant faults) change a little mainly because the test circuit with the reconfigured scan forest is different from the fully scanned circuit with a scan chain. The CPU time presents the time to establish the reconfigured scan forest. The test data volume for b19 can be compressed to 2.78 percent of that of the single scan chain designed circuit.

6 CONCLUSIONS

A new scan architecture, called the reconfigured scan forest, was proposed for low test application cost, reduced test data volume, and low test power consumption. The reconfigured scan forest can greatly reduce the area overhead and the routing complexity of the XOR trees for test response compaction. A new procedure was proposed to construct the reconfigured scan forest and the XOR trees concurrently, which avoids aliasing faults. It is found that the reconfigured scan forest can be constructed in very short time for all circuits used. Experimental results were presented to compare with a number of recent methods on test application cost, test data volume, and test power.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation of China under grants 60373009 and 60425203 and JSPS under grant L03540.

REFERENCES

- [1] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design*. IEEE Press, 1994.
- [2] I. Bayraktaroglu and A. Orailoglu, "Concurrent Application of Compaction and Compression for Test Time and Data Volume Reduction in Scan Designs," *IEEE Trans. Computers*, vol. 52, no. 8, pp. 1480-1489, Aug. 2003.
- [3] B.B. Bhattacharya, S.C. Seth, and S. Zhang, "Double-Tree Scan: A Novel Low-Power Scan-Path Architecture," *Proc. IEEE Int'l Test Conf.*, pp. 470-479, 2003.
- [4] Y. Bonhomme, T. Yoneda, H. Fujiwara, and P. Girard, "An Efficient Scan Tree Design for Test Time Reduction," *Proc. Ninth IEEE European Test Symp.*, pp. 321-326, 2004.
- [5] M.L. Bushnell and V.D. Agrawal, *Essentials of Electronic Testing*. Kluwer Academic, 2000.
- [6] K. Chakrabarty and B.T. Murray, "Design of Built-In Test Generator Circuits Using Width Compression," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 10, pp. 1044-1051, 1998.
- [7] A. Chandra and K. Chakrabarty, "Low-Power Scan Testing and Test Data Compression for Systems-on-a-Chip," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 5, pp. 597-604, 2002.
- [8] A. Chandra and K. Chakrabarty, "Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes," *IEEE Trans. Computers*, vol. 52, no. 8, pp. 1076-1088, Aug. 2003.
- [9] J.J. Chen, C.K. Yang, and K.J. Lee, "Test Pattern Generation and Clock Disabling for Simultaneous Test Time and Power Reduction," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 3, pp. 363-370, 2003.
- [10] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S.M. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 12, pp. 1325-1333, 1998.
- [11] E.B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability," *Proc. ACM/IEEE Design Automation Conf.*, pp. 462-468, 1977.
- [12] I. Hamzaoglu and J.H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," *Proc. IEEE Int'l Symp. Fault-Tolerant Computing*, pp. 260-267, 1999.
- [13] S. Gerstendofer and H.J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST," *Proc. IEEE Int'l Test Conf.*, pp. 77-84, 1999.
- [14] H.K. Lee and D.S. Ha, "On the Generation of Test Patterns for Combinational Circuits," Technical Report 12-93, Dept. of Electrical Eng., Virginia Polytechnic Inst. and State Univ., 1993.
- [15] K.J. Lee, J.J. Chen, and C.H. Huang, "Using a Single Input to Support Multiple Scan Chains," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design*, pp. 74-78, 1998.
- [16] S.Y. Lee and K.K. Saluja, "Test Application Time Reduction for Sequential Circuits with Scan," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 9, pp. 1128-1140, 1995.
- [17] S. Mitra and K.S. Kim, "x-Compact: An Efficient Response Compaction Technique for Test Cost Reduction," *Proc. IEEE Int'l Test Conf.*, pp. 311-320, 2002.
- [18] K. Miyase and S. Kajihara, "Optimal Scan Tree Construction with Test Vector Modification for Test Compression," *Proc. 12th IEEE Asian Test Symp.*, pp. 136-141, 2003.
- [19] N. Nicolici and B.M. Al-Hashimi, "Multiple Scan Chains for Power Minimization during Testing Application in Sequential Circuits," *IEEE Trans. Computers*, vol. 51, no. 6, pp. 721-734, June 2002.
- [20] I. Pomeranz and S.M. Reddy, "Test Data Compression Based on Input-Output Dependence," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 10, pp. 1450-1455, 2003.
- [21] D.K. Pradhan and J. Saxena, "A Novel Scheme to Reduce Test Application Time in Circuits with Full Scan," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 12, pp. 1577-1586, 1995.
- [22] S. Wang and S. Gupta, "ATPG for Heat Dissipation Minimization during Test Application," *IEEE Trans. Computers*, vol. 47, no. 2, pp. 256-262, Feb. 1998.
- [23] M.J.Y. Williams and J.B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Trans. Computers*, vol. 22, pp. 46-60, 1973.
- [24] P. Wohl, J.A. Waicukauski, and T.W. Williams, "Design of Compactors for Signature-Analyzers in Built-In Self-Test," *Proc. IEEE Int'l Test Conf.*, pp. 54-63, 2001.
- [25] D. Xiang, S. Gu, J.G. Sun, and Y.L. Wu, "A Cost-Effective Scan Architecture for Scan Testing with Nonscan Test Power and Test Application Cost," *Proc. ACM/IEEE Design Automation Conf.*, pp. 744-747, June 2003.
- [26] D. Xiang, Y. Xu, and H. Fujiwara, "Non-Scan Design for Testability for Synchronous Sequential Circuits Based on Conflict Resolution," *IEEE Trans. Computers*, vol. 52, no. 8, pp. 1063-1075, Aug. 2003.
- [27] D. Xiang and J.H. Patel, "Partial Scan Design Based on Circuit State Information and Functional Analysis," *IEEE Trans. Computers*, vol. 53, no. 3, pp. 276-287, Mar. 2004.
- [28] H. Yotsuyanagi, T. Kuchii, S. Nishikawa, M. Hashizume, and K. Kinoshita, "Reducing Scan Shifts Using Folding Scan Trees," *Proc. 12th IEEE Asian Test Symp.*, pp. 6-11, 2003.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.