11th IEEE European Test Symposium, pp. 13-18, May 2006.

Extended Compatibilities for Scan Tree Construction

Zhiqiang You Michiko Inoue Hideo Fujiwara
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
{you-z, kounoe, fujiwara}@is.naist.jp

Abstract

Scan tree techniques reduce test application time significantly by shifting test values into (out from) the compatible flip-flops simultaneously. This paper proposes a novel scan tree architecture for test application time and test power reduction. In this proposed method, the compatibility is extended by employing NOT gates and XOR gates. Experimental results show that our approach is more effective to achieve short test application time and low test power compared with the conventional scan tree design.

Key words: design for testability, full scan testing, scan tree, low power testing

1. Introduction

With the transistor counts exponential increase, scan-based designs are widely employed to reduce test generation time. Full scan-based design is one of the most important design for testability (DFT) methodologies in very large scale integration (VLSI) circuits and in system-on-hip (SoC) cores. In this DFT methodology, all flip-flops are enhanced to scan cells, and test application time depends on the length of the longest scan chain. Though full scan design reduces test generation complexity drastically, the test cost is very high that increases the cost of automatic test equipment (ATE).

There huge number techniques have investigated low cost test. Some methodologies [1-4] explore new scan architectures. The method in [1] effectively reduces test data volume and test application time for designs with multiple scan chains using a reconfigurable switch to apply tests from a limited number of external inputs to a large number of internal scan chains. VirtualScan technology is proposed in [2] to reduce test cost based on the idea of reducing the longest scan chain length in a full-scan circuit. The technique in [3] reduces test data volume and test application time drastically by employing the CircularScan architecture that uses the captured response for the next vector by replacing only necessary bits.

However, in these methods, to achieve short test application time there are too many transitions in the circuit under test (CUT). The power dissipation is quite high. If the test power dissipation exceeds the designed power constraint, it can give rise to severe hazards in circuit reliability or can provoke instant circuit damage [4]. Hence, it is more important to achieve low test cost with low test power.

Various techniques have been proposed to reduce switching activity during test to reduce power. The methodologies in [5-8] employ test vector or scan cell reordering technique where test vectors in a test set or scan cells for a test set are reordered for minimal power consumption. The methodologies in [9,10] also explore the correlation between consecutive test patterns by filling each don't care bit in the test cubes with appropriate value 0 or 1. There are some methods [11-15] that reduce power consumption by using scan chain disabling technique. In methods of [11-14], only one scan chain at the same time is activated during scan shifting. The power during scan shifting is reduced to $1/N$, where N is the number of scan chains. [15] reduces both peak power and average power dissipation by simultaneously activating only one scan chain during both shift and cycles. However, these methodologies did not consider test application time or test data volume reduction.

Recently, scan tree techniques [16-20] have been proposed to reduce test application time and test power. In these techniques, scan cells are constructed into a tree structure. The length of the longest scan chain is reduced. During scan operation, test data are shifted into the scan tree via one scan cell at the root. The scan cells in the same level have the same shifted test data. Therefore, to keep fault coverage, the scan cells should be compatible for all the test vectors.

As the first technique concerning scan tree, [16] constructs a scan tree and minimizes its height by test vector modification. [17] extends the methodology in [16] into multiple scan chain cases. [18] improves the solution of scan tree technique by adopting a folding mode to enhance the parallelism. In [20], test application time and test data volume are reduced drastically by a dynamic reconfiguration between the scan chain mode and the single scan mode.

In this paper, we extend the concept of compatibility for scan tree techniques. And we propose a novel scan tree architecture to achieve short test application time and low test power. Notice that, the method in [20] adapts to our methodology. If we apply this technique to our method, the test application time can be reduced more and our technique is applicable to update test vectors for the CUT.

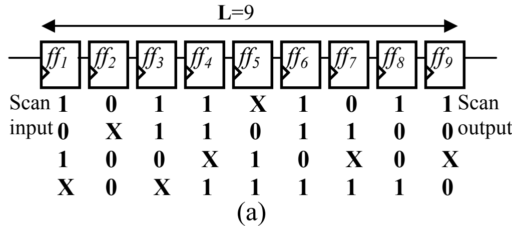
This paper is organized as follows. Section 2 introduces some basic concepts about scan tree techniques. Section 3 describes our proposed scan tree architecture with extended compatibilities. Section 4 gives the algorithm to construct a scan tree. Section 5

reports on some experimental results for our proposed method. Section 6 concludes with a brief summary.

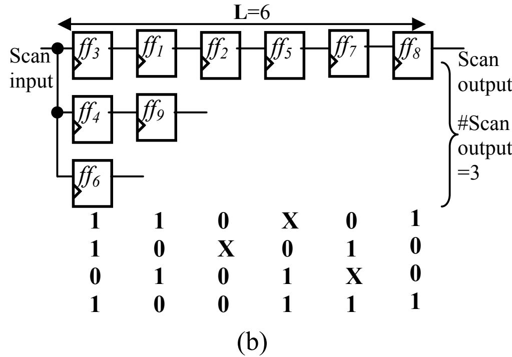
2. Scan Tree Architecture

Scan tree architectures group the compatible scan flip-flops in the same level so that the scan cells can receive the same test values for all the test vectors. These technologies shorten the length of the scan chain, and hence, reduce test application time and test data volume. Scan tree methodologies are more effective for the test set with don't care values. Let X denote a don't care value.

Definition 1. For a test set T, scan cells ff_i and ff_j are *normal-compatible*, if for any test cube c_m in T, $v_{i,m} = v_{j,m}$, $v_{i,m}$ = X or $v_{j,m}$ = X, where $v_{i,m}$ is the value of scan cell i of test cube c_m.



(a)



(b)

Fig. 1. Scan tree architecture

Fig. 1.(a). shows an example of a single scan chain with 9 scan cells and a test set. According the compatibility of scan cells, ff_3, ff_4 and ff_6, ff_1 and ff_9 are grouped in the same level respectively. ff_2, ff_5, ff_7, ff_8 are not compatible with any other scan cells. After the scan cells are grouped, some don't care bits corresponding to the scan cells may be specified. The scan tree architecture and its test set are shown in Fig. 1.(b). The X in ff_1 is specified to 0 since value of the corresponding bit in its compatible scan cell, ff_9 is 0. The length of the longest scan chain is reduced from 9 to 6. Therefore, both test application time and test data volume are reduced by 1/3. The number of scan outputs is 3. We can use an embedded MISR to analyze the test responses.

3. Proposed Scan Tree Architecture

In this section, we will describe the proposed scan tree architecture.

3.1 Compatibility Extension

We give a basic definition and extend a concept of normal-compatibility by employing NOT gates and XOR gates in this subsection.

Definition 2. A scan cells is in the i-th *level* of a scan tree, if its test data can be shifted into it from the scan input through i scan shift cycles.

Definition 3. For a test set T, scan cells ff_i and ff_j are *NOT-compatible*, if for any test cube c_m in T, $v_{i,m} \neq v_{j,m}$, $v_{i,m}$ = X or $v_{j,m}$ = X, where $v_{i,m}$ is the value of scan cell i of test cube c_m.

Definition 4. For a test set T, scan cells ff_i and ff_j are *XOR-compatible* for ff_k, if for any test cube c_m in T, $v_{i,m} \oplus v_{j,m} = v_{k,m}$, $v_{i,m}$ = X, $v_{j,m}$ = X or $v_{k,m}$ = X, where $v_{i,m}$ is the value of scan cell i of test cube c_m.

Using both NOT gate and XOR gate, the *NXOR-compatibility* is defined as follows.

Definition 5. For a test set T, scan cells ff_i and ff_j are *NXOR-compatible* for ff_k, if for any test cube c_m in T, $v_{i,m} \oplus \mathrm{NOT}(v_{j,m}) = v_{k,m}$, $v_{i,m}$ = X, $v_{j,m}$ = X or $v_{k,m}$ = X, where $v_{i,m}$ is the value of scan cell i of test cube c_m.

To simplify the notation, we will use ff_i = NOT (ff_j), $ff_k = ff_i \oplus ff_j$, $ff_k = ff_i \oplus \mathrm{NOT}(ff_j)$ to represent the above compatibilities respectively.

3.2 Proposed Architecture

The scan cells that are normal- or NOT-compatible with each other can be grouped into a clique. We describe three theorems that are the base to construct a scan tree of the extended compatibilities as follows.

Theorem 1. The scan cells in a clique can be grouped into the same level.

Theorem 2. If there exist two cliques C_i and C_j, and scan cells ff_k, ff_m are XOR- or NXOR-compatible with any scan cell in C_i for any scan cell in C_j respectively, the following statement is true. The scan cells ff_k, ff_m are normal- or NOT-compatible if and only if it does not cause any conflict when specifying don't care bits corresponding to the scan cells in C_i and C_j.

We can find the maximal clique C_k where any scan cell is XOR- or NXOR-compatible with any scan cell in C_i for any scan cell in C_j. To simplify, we call this case as C_k is XOR-compatible with C_i for C_j.

Theorem 3. If there exist three cliques C_i, C_j and C_k, and any scan cell in C_k are XOR- or NXOR-compatible with any scan cell in C_i for any scan cell in C_j, the scan cells of C_k can be grouped in the same level as the higher level between C_i and C_j in a scan tree.

Theorem 4. Among the cliques in level l_1, there exists only one clique $C_{l1,0}$ so that for any other clique $C_{l1,m1}$ (m_1>0) there exist a clique $C_{l2,m2}$ (l_2<l_1, m_2≥0) $C_{l1,m1}$ is XOR-compatible with $C_{l1,0}$ for $C_{l2,m2}$.

We call the clique $C_{l1,0}$ as *primary clique* in level l_1.

The scan tree is constructed as following. First, we connect the scan cells in the primary cliques between the neighboring levels. Secondly, we connect

the scan cells in other cliques according to their XOR- or NXOR-compatibilities. Notice that, to keep fault coverage, in the second step, we cannot remove the scan output of step 1as one input of an XOR gate. And all the scan cells are not in the primary cliques have scan output themselves.

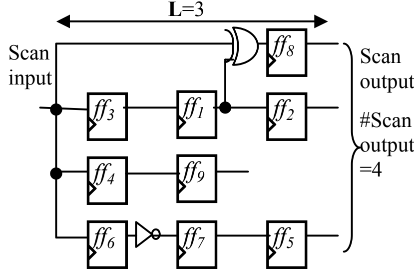


Fig. 2. Scan tree architecture with the extended compatibilities

Fig. 2. gives a scan tree architecture with the extended compatibilities for the scan cells and the test set shown in Fig. 1.(a). Since ff_7 is NOT-compatible with ff_1 and ff_9, they are grouped into the second level. ff_2 and ff_5 are grouped into a clique because they are NOT-compatible. ff_2, ff_8 and ff_5, ff_8 are XOR-compatible for any scan cell in the first level. Therefore, we can put ff_2, ff_5, ff_8 into the third level. The construct scan tree is shown in Fig. 2. The length of the longest scan chain is reduced to 3. Test application time and test data volume are reduced by 2/3. The added hardware is one NOT gate and one XOR gate. The number of scan outputs is 4, that is, a little higher than the scan tree architecture in Fig. 1.(b)..

3.3 The Number of Scan Outputs Reduction

The higher number of scan outputs may cause higher hardware overhead to analyze the test responses of scan cells. To reduce the number of scan outputs without affecting fault coverage, we compact scan outputs according to the following three rules. In the following rules, to simplify we show and explain them to employ scan cells. It also effect by change the term from scan cell to clique.

Rule 1. If a scan cell ff_i is XOR- or NXOR-compatible with scan cell ff_j for the scan cell ff_k of the first level, and there exists one scan output O_l in the preceding level, the scan output can be connected to one input of the XOR gate to reduce one scan output. This is because the test responses of scan cells along the scan path from scan input to the scan output O_l cannot be masked when do scan operations.

For example, in Fig. 2, ff_8 is such scan cell. There

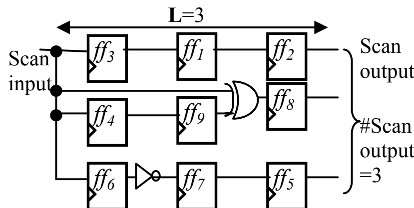


Fig. 3. Reducing the number of scan output using Rule 1.

is a scan output at the second level. We utilize this line to propagate test data for ff_8. The result is shown in Fig. 3.

Rule 2. If scan cells ff_{i1}, ff_{i2} are XOR- or NXOR-compatible with scan cells ff_{j1}, ff_{j2} for the scan cells ff_{k1}, ff_{k2} respectively, and ff_{k1}, ff_{j1} are direct predecessors of ff_{k2}, ff_{j2} respectively, the scan cell ff_{i2} can be a direct successor of ff_{i1}. The reason is as follows. When performing scan operation, the test data of ff_{j2}, ff_{k2} are propagated from ff_{j1}, ff_{k1}. Just before the last shift cycle, ff_{j1}, ff_{k1} store the test data of ff_{j2}, ff_{k2}. Because $ff_{i1} = ff_{j1} \oplus ff_{k1}$ ff_{i1} stores the test data of $ff_{j2} \oplus ff_{k2}$. Therefore, ff_{i2} can be the next scan cell of ff_{i1} along the scan tree.

For example, in Fig. 4.(a), scan cells ff_4, ff_5 have the same situation with ff_{i1} and ff_{i2}. $ff_4 = ff_1 \oplus ff_2$, while $ff_5 = ff_2 \oplus ff_3$; ff_1 and ff_2 are the scan cells just before ff_2

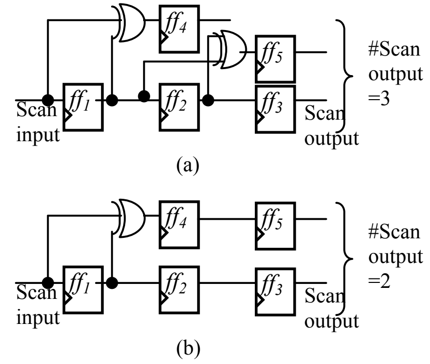


Fig. 4. Reducing the number of scan output using Rule 2.

and ff_3 in the scan tree respectively. After removing a scan output, the result is shown in Fig. 4.(b).

Rule 3. If a scan cell ff_j has more than one outputs where one is a scan output O_l, and if there exists one of their direct successors of gates or scan cells e_h where the test response of all the direct

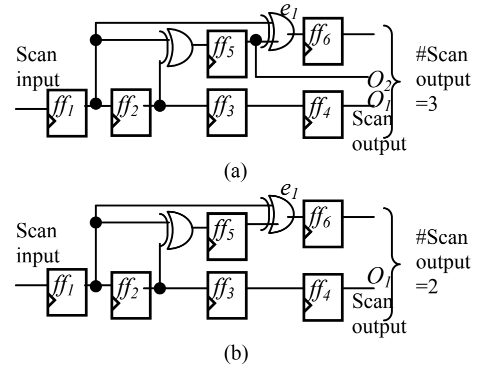


Fig. 5. Reducing the number of scan output using Rule 3

predecessors of scan cells except ff_j can be propagated a scan output without e_h, the scan output O_l can be removed. This is because the test response of ff_j can be propagated through e_h to a scan output.

For example, as shown in Fig. 5.(a), ff_5 has two outputs, and one of the outputs O_2 is a scan output. e_1 is its a direct successor. There is one predecessor ff_1 of e_1 except ff_5. The test response of ff_1 can be to a scan

output O_2. Fig. 5.(b). shows the scan tree after the reduction.

In the following approach, we realize these rules to reduce the number of scan outputs. There are still some rooms to reduce more. For instance, if we consider the structure of a circuit, which different scan paths may share one scan output if the scan cells in the same level from output cannot capture error of a fault, the number of scan outputs can be reduced. Data compaction techniques [21], also give solutions to reduce the number of scan outputs. In this paper, we do not deal with these cases.

3.4 Average Power Reduction

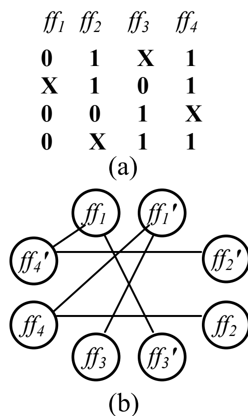
Our method probably reduces the transitions between the neighboring levels of a scan tree to employed NOT gates and XOR gates. Therefore, the proposed architecture adapts to reduce average power. The results shown in the experimental results part will evaluate the efficiency.

4. Optimal Algorithm to Construct a Scan Tree

In this section, we describe the proposed approach to find an optimal scan tree for full scan design with one scan input and a given test set.

4.1 Compatibility Graph

A compatibility graph represents the relations of scan cells. To explain the NOT-compatibility, a scan cell is represented by two nodes. One node represents the scan cell itself. The other node stands for its complement values. An edge exists between two nodes if the corresponding scan cells are normal-compatible. For example, for the given test cubes in Fig. 6.(a), the compatibility graph can be constructed as Fig. 6.(b).



ff_1	ff_2	ff_3	ff_4
0	**1**	**X**	**1**
X	**1**	**0**	**1**
0	**0**	**1**	**X**
0	**X**	**1**	**1**

(a)

(b)

Fig. 6. Compatibility graph

4.2 Overview

The aim of this algorithm is to find the minimum number of levels of cliques. Since the problem is NP-hard, we use a heuristic algorithm shown in Fig. 7. We first construct the compatibility graph G with normal- and NOT-compatibilities (line 1). In this algorithm, we reduce a clique into a node. To distinguish the original nodes, in the rest we call the node as a clique node while the original nodes are named as scan cell nodes. During the scan tree construction process (lines 3-7), we find a clique set $\mathbb{C}_i$ of G such that the number of scan cells that are in all the cliques of clique set $\mathbb{C}_i$ are maximized (line 4, algorithm FINDCLIQUES shown in section 4.3). Algorithm FINDCLIQUES also performs to specify don't care bits of a test set for the compatible scan cells, to update the graph G, and to generate the i-th level of the scan tree. The algorithm ends until graph G is empty.

```
Scan tree construction algorithm for extended compatibilities
1. Construct the compatibility graph G;
2. i=1;
3. Repeat {
4.   FINDCLIQUES ℂi of G;
7.   i++;
8. }
9. Until G is empty;
```

Fig. 7. Scan tree construction algorithm

4.3 Finding a Maximal Clique Set

In this subsection, we will give an algorithm to find a maximal clique set $\mathbb{C}$ of G. This problem is high complexity. We employ a heuristic algorithm, shown in Fig. 8, where we first find a maximal clique $C_{i,0}$ (line 1). Secondly, we specify some don't care bits of test values corresponding to the scan cells represented by the nodes in the clique $C_{i,0}$ (line 2). Thirdly, we record a node $p_{i,0}$ to present the clique (line 3). Fourthly, we update graph G by removing all the scan cell nodes of the clique and their complement nodes (line 4). After that, for every clique node $p_{j,k}$ in G, where $j<i$, we find the maximal clique $C_{i,m}$ where the nodes are XOR- or NXOR-compatible with $p_{i,0}$ for $p_{j,k}$. Then, Specify some don't care bits of the test set corresponding to the nodes in $C_{i,m}$, $C_{i,0}$ and $C_{j,k}$ considering not only the compatibilities of inside the cliques $C_{i,m}$, but also the relation of $C_{i,m}$, $C_{i,0}$ and $C_{j,k}$. Here, the values of some bit of $C_{i,m}$, $C_{i,0}$ and $C_{j,k}$ may be "partially" specified. For example, if one bit of the values of three XOR compatible nodes in a test cube are X, X, and 0 respectively. The relation is not enough to specify the don't care bit. Nevertheless, the first two Xs should have the same specified values when applying the test to a CUT. We keep such information for the further specification. We record a node $p_{i,m}$ to present the clique And graph G is updated in the same way as the beginning of the paragraph. According to theorem 3, all the cliques in a clique set can be grouped in the same level. Next, the i-th level of the scan tree is generated by connecting the scan cells in the (i-1)-th level (line 13). In this step, the connection between the scan cells in the neighboring levels are considered to reduce hardware overhead by sharing NOT gates, XOR gates

Algorithm FINDCLIQUES: Finding the maximal clique set $\mathbb{C}_i$

1. Find a maximal clique $C_{i,0}$;
2. Specify some don't care bits of the test set corresponding to the nodes in the clique;
3. Record a node $p_{i,0}$ to present the clique;
4. Remove the nodes in $C_{i,0}$ and their complement nodes;
5. m=1;
6. For every clique node $p_{j,k}$ (except $p_{i,0}$) {
7. If existing, find the maximal clique $C_{i,m}$ where the nodes are XOR- or NXOR-compatible with $p_{i,0}$ for $p_{j,k}$;
8. Specify some don't care bits of the test set corresponding to the nodes in $C_{i,m}$, $C_{i,0}$ and $C_{j,k}$;
9. Record a node $p_{i,m}$ to present the clique;
10. Remove the nodes in $C_{i,m}$ and their complement nodes;
11. m++;
12. }
13. Generate the i-th level of the scan tree by connecting the scan cells in the (i-1)-th level;
14. Return the clique set $\mathbb{C}_i$ and the i-th level of the scan tree;

Fig. 8. Finding the maximal clique set

or the rules of scan output reduction. Finally, the clique set $\mathbb{C}_i$ and the i-th level of the scan tree are returned (line 14).

Notice that, in our method we group the large clique set in the low level of a scan tree, which efficiently explores the compatibilities and reduce hardware overhead.

5. Experimental Results

We have conducted experiments on full scan version of ISCAS'89 benchmark circuits in C language on a Pentium III Mobile 800MHz with 256 MB RAM. In the experiments, we use the ATPG tool "TestGen" of Synopsys to generate test cubes and perform fault simulation.

Table 1 shows the results of ISCAS'89 benchmark circuits. The first four columns give the circuit's names, the number of scan cells, the number of test vectors and fault coverage. The following three columns draw the percentages of shift time or test data reduction between the scan tree technique with normal-compatibility, the scan tree technique with normal- and NOT-compatibilities, the proposed method and that of the single scan chain respectively. The added number of NOT gates, XOR gates and scan outputs are shown in the next three columns. The column "Av. Red. (%)" reports the percentages of average power reduction of the proposed method compared with full scan design with one scan chain. The last column displays the computation time to construct a scan tree and to obtain its test vectors in seconds.

In this experiment, we estimate shift time reduction by the longest scan path reduction. As some published methods do, we use the technique in [9] to estimate test power.

As shown in Table 1, the shift time reduction for the scan tree technique with normal-compatibility is 51.2% in average, and up to 87.2%. The method employing one extended compatibility, NOT-compatibility can achieve a little better results. The reduction is 54.4% in average, and up to 88.4%. Our proposed method is more effective to reduce test application time and test data. The reduction is 68.9% in average, and up to 97.9%. The hardware overhead is not very high. The computation time is short even for the largest circuits.

Notice that, the average power dissipation also can be reduced in the architecture to reduce test application time. The percentage of the reduction is 64.4 in average. The proposed method also adapts to reduce test power.

6. Conclusions

This paper proposed a novel scan tree architecture for test application time, test data volume and test power reduction. In this architecture, the compatibility is extended to NOT-, XOR-, NXOR- and normal-compatibilities by employing NOT gates and XOR gates. Experimental results show that our approach is more effective to achieve short test application time, low test data volume compared with the conventional scan tree design. The average power is also reduced drastically.

The scan chain routing problem may occur if some scan cells that have far distances are connected together. Future work will investigate the layout impact to our methodology.

Acknowledgments

This work was supported in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research B(2) (No. 15300018).

The authors wish to thank Drs. Satoshi Ohtake, Tomokazu Yoneda and Thomas Clouqueur of Nara Institute of Science and Technology, for their valuable comments. Thanks are also due to the members of Fujiwara laboratory.

References

[1] H. Tang, S. M. Reddy and I. Pomeranz, "On reducing test data volume and test application time for multiple scan chain designs," In Proc. IEEE International Test Conference, pp. 1079-1088, 2003.

[2] L.-T. Wang, X. Wen, H. Furukawa, F.-S. Hsu, S.-H. Lin, S.-W. Tsai, K. S. Abdel-Hafez and S. Wu, "VirtualScan: a new compressed scan Technology for test cost

reduction", In Proc. IEEE International Test Conference, pp. 916-925, 2004.

[3] B. Arsian and A. Orailoglu, "CircularScan: a scan architecture for test cost reduction", In Proc. Of the IEEE Design, Automation and Test in Europe Conference, pp. 1290-1295, 2004.

[4] Patrick Girard, "Survey of Low-Power Testing of VLSI Circuits," IEEE Design & Test of Computers, vol. 19, no. 3, pp82-92, 2002.

[5] S. Chakravarty and V. Dabholkar, "Two techniques for minimizing power dissipation in scan circuits during test application," In Proc. IEEE Asian Test Symposium, pp. 324-329, 1994.

[6] 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. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 12, pp. 1325-1333, 1998.

[7] P. Flores. J. Costa, H. Neto, J. Monterio and J. Marq1ues-Silva, "Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation," Proc. Of International Conference on VLSI Design, pp. 37-41, 1999.

[8] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "Efficient scan chain design for power minimization during scan testing under routing constraint," In Proc. IEEE International Test Conference, pp. 488-493, 2003.

[9] R. Sankaralingam, R. R. Oruganti and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," In Proc. IEEE VLSI Test Symposium, pp. 35-40, 2000.

[10] K. M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain and J. Lewis, "Minimizing power consumption in scan testing: pattern generation and DFT techniques," In Proc. IEEE International Test Conference, pp. 355-364, 2004.

[11] L.Whetsel, "Adapting scan architectures for low power operation," In Proc. IEEE International Test Conference, pp. 863-872, 2000.

[12] J. Saxena, K. M. Butler and L. Whetsel, "An analysis of power reduction techniques in scan testing," In Proc. IEEE International Test Conference, pp. 670-677, 2001.

[13] Y. Bonhomme, P. Girard, L. Guiller C. Landrault and S. Pravossoudovitch, "A gated clock scheme for low power scan testing of logic Ics or embedded cores," In Proc. IEEE Asian Test Symposium, pp. 253-258, 2001.

[14] B. B. Bhattacharya, S. C. Seth and S. Zhang, "Double-tree scan: a novel low-power scan-path architecture," In Proc. IEEE International Test Conference, pp. 471-479, 2003.

[15] Z. You, T. Iwagaki, M. Inoue and H. Fujiwara, "A low power deterministic test using scan chain disable technique," IEEE 6th Workshop on RTL and High Level Testing, pp.184-191, July 2005.

[16] K. Miyase and S. Kajihara, "Optimal Scan Tree Construction with Test Vector Modification for Test Compression," In Proc. IEEE Asian Test Symposium, pp. 136-141, 2003.

[17] K. Miyase, S. Kajihara and S. M. Reddy, "Multiple Scan Tree Design with Test Vector Modification," In Proc. IEEE Asian Test Symposium, pp. 76-81, 2004.

[18] H. Yotsuyanagi, T. Kuchii, S. Nishikawa, M. Hashizume, K. Kinoshita, "Reduce Scan Shifts using Folding Scan Trees," In Proc. IEEE Asian Test Symposium, pp. 6-11, 2003.

[19] D. Xiang, S. Gu, J.G. Sun and Y. Wu, "A Cost-Effective Scan Architecture for Scan Testing with Non-scan Test power and Test application Cost," In Proc. Design Automation Conference, pp. 744-747, 2003.

[20] Y. Bonhomme, T. Yoneda, H. Fujiwara and P. Girard, "An Efficient Scan Tree Design for Test Time Reduction," In Proc. IEEE European Test Symposium, pp. 174-179, 2004.

[21] T. Clouqueur, K. Zarrineh, K. K. Saluja and Hideo Fujiwara, "Design and Analysis of Multiple Weight Linear Compactors of Responses Containing Unknown Values," In Proc. IEEE International Test Conference, 2005.

Table 1. Shift time and average power reduction of the proposed scan tree architecture

Benchmark circuits	#FFs	#test vect.	FC (%)	Shift time red.(%)			Hardware			Av. Red. (%)	Comput. time(sec.)
				tree/scan	not/scan	xor/scan	not	xor	sout		
S1423	74	88	98.99	20.3	25.7	59.5	4	24	5	51.6	0.1
S5378	179	162	99.05	32.4	34.6	45.3	11	26	17	34.0	0.8
S9234	211	348	93.16	30.8	34.1	49.8	7	43	22	43.6	1.7
S13207	699	80	98.32	66.1	69.2	79.3	33	95	137	77.1	16.7
S15850	597	351	96.37	58.5	61.6	74.9	53	105	109	72.5	12.3
S35932	1728	2879	88.61	87.2	88.4	97.9	249	298	273	97.5	183.2
S38417	1636	651	99.39	64.1	66.6	78.4	109	376	251	77.4	254.8
S38584	1452	1056	95.00	51.5	54.6	66.3	131	348	54	61.7	211.1
Average	-	-	-	51.4	54.4	69	75	175	109	64.4	85.1