Test Compaction for Transition Faults under Transparent-Scan

Irith Pomeranz[1] and Sudhakar M. Reddy[2]
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907, U.S.A.

Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242, U.S.A

Abstract

Transparent-scan was proposed as an approach to test generation and test compaction for scan circuits. Its effectiveness was demonstrated earlier in reducing the test application time for stuck-at faults. We show that similar advantages exist when considering transition faults. We first show that a test sequence under the transparent-scan approach can imitate the application of broadside tests for transition faults. Test compaction can proceed similar to stuck-at faults by omitting test vectors from the test sequence. A new approach for enhancing test compaction is also described, whereby additional broadside tests are embedded in the transparent-scan sequence without increasing its length or reducing its fault coverage.

1. Introduction

The *transparent−scan* approach was proposed in [1]. By eliminating the distinction between scan operations and functional clock cycles during test generation and test compaction, the transparent-scan approach reduces the number of clock cycles required for testing the circuit. A test under the transparent-scan approach is a sequence T of primary input vectors. A primary input vector assigns values to the primary inputs of the original circuit (the circuit without scan), to the scan chain input(s), and to the scan select input. An output vector includes values corresponding to the primary outputs of the original circuit, and to the scan chain output(s). This unified view of the primary inputs (outputs) of the original circuit and the scan inputs (outputs) provides complete flexibility in interleaving scan clock cycles and functional clock cycles. It naturally results in *limited scan operations*, where a scan chain is shifted a number of times smaller than its length [2]-[8]. The transparent-scan approach was also found to be necessary in [9] for testing critical paths in a microprocessor that uses partial scan. Test application under transparent-scan is similar to test application for a synchronous sequential circuit without scan.

In this work we investigate the effectiveness of the transparent-scan approach in reducing the test application time when the target faults are delay faults. We consider transition faults. We use for transition faults tests that are equivalent to broadside tests [10]. A broadside test $t = <SI, A_1, A_2>$ starts by scanning in a state denoted by SI. A primary input vector denoted by A_1 is then applied in functional mode under a slow capture cycle to initialize the circuit lines to known values. The next state obtained under SI and A_1 is latched in the flip-flops. A primary input vector A_2 is then applied in functional mode under a fast capture cycle to create signal-transitions, and propagate fault effects to the flip-flops. The next state obtained after the application of A_2 is latched in the flip-flops, and then scanned out as the test response.

Transparent-scan can imitate the application of broadside tests as shown in Section 2. This is achieved by a method similar to the test set translation method from [1], where a scan-based stuck-at test set for the circuit is translated into a transparent-scan test sequence. The structure of the transparent-scan sequence is different when broadside tests are translated. In addition, we associate with every vector of the transparent-scan test sequence a flag indicating whether the vector is applied under a slow or a fast capture cycle. This flag is not needed when stuck-at faults are considered.

Transparent-scan provides several opportunities for the application of test compaction methods to transition faults. Some of these methods are extensions of methods considered for stuck-at faults in [1]. Others are unique to delay faults. These methods are explored in Sections 3, 4, 5 and 6. The first compaction method, explored in Section 3, is based on the omission of test vectors from the transparent-scan sequence without reducing the transition fault coverage. This is similar to the test compaction process applied for stuck-at faults in [1]. However, care must be taken to ensure that the test sequence after compaction preserves the structure of broadside tests embedded in it.

The second compaction method, explored in Section 4, is unique to delay faults. It consists of replacing slow capture cycles with fast capture cycles, and scan clock cycles with functional clock cycles, in order to increase the number of broadside tests embedded in the transparent-scan sequence. In this way, the number of clock cycles where faults may be activated and eventually detected is increased. After the embedding of additional broadside tests, the vector omission process is able to omit additional test vectors from the sequence, thus further reducing its test application time. A similar process can

1. Research supported in part by SRC Grant No. 2004-TJ-1244.
2. Research supported in part by SRC Grant No. 2004-TJ-1243.

also be used for embedding other types of tests for transition faults in the transparent-scan sequence.

We describe the overall compaction process for transition faults under transparent-scan in Section 5 and present experimental results in Section 6.

2. Test set translation for broadside test sets

The test set translation process described in this section accepts a standard scan-based broadside test set B and translates it into a transparent-scan sequence T. The test set B and the transparent-scan sequence T are equivalent in the sense that they apply exactly the same tests to the circuit. They also require the same number of clock cycles, and have the same sequences of scan and functional clock cycles, and the same sequences of slow and fast capture cycles.

We denote the original primary inputs of the circuit by $a_0, a_1, \cdots, a_{n-1}$. We assume that the circuit has a single scan chain with a scan select input s_{sel}, and a scan chain input s_{inp}. Under transparent-scan, a vector of a test sequence is defined over the inputs $a_0 a_1 \cdots a_{n-1} s_{sel} s_{inp}$, in this order. A functional clock cycle u has $s_{sel}(u) = 0$, and a scan clock cycle has $s_{sel}(u) = 1$. We use a variable ρ to indicate whether a slow or fast capture cycle is used for a given time unit u. We have $\rho(u) = 1$ for a time unit u where a fast capture cycle is used, and $\rho(u) = 0$ for a time unit u where a slow capture cycle is used.

We assume that the scan chain is shifted to the right. As a result, to scan in the state $SI_i = SI_i^0 \cdots SI_i^{k-1}$ starting at time unit u_0, we must hold $s_{sel}(u) = 1$ at time units $u = u_0, u_0+1, \cdots, u_0+k-1$, and set $s_{inp}(u_0) = SI_i^{k-1}$, $s_{inp}(u_0+1) = SI_i^{k-2}$, $\cdots$, $s_{inp}(u_0+k-1) = SI_i^0$. For example, to scan in the state 011 into the scan chain of Figure 1 starting at time unit u_0 we must hold $s_{sel}(u) = 1$ for $u = u_0, u_0+1, u_0+2$, and we must set $s_{inp}(u_0) = 1$, $s_{inp}(u_0+1) = 1$ and $s_{inp}(u_0+2) = 0$.

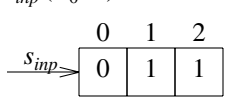

Figure 1: Scanning in a state

We demonstrate the test set translation process by considering a broadside test set for ISCAS-89 benchmark circuit $s27$. The original circuit has four primary inputs and three state variables. A broadside test set for the circuit is shown in Table 1. The scan in state of test t_i is denoted by SI_i, and the primary input vectors of test t_i (defined over the primary inputs of the original circuit) are denoted by A_{i1} and A_{i2}.

To apply $t_0 = <SI_0, A_{01}, A_{02}>$ under transparent-scan, we must first scan in $SI_0 = 001$. This is accomplished by setting $s_{sel}(0) = 1$, $s_{sel}(1) = 1$, $s_{sel}(2) = 1$, $s_{inp}(0) = 1$, $s_{inp}(1) = 0$ and $s_{inp}(2) = 0$. The values of the original primary inputs are don't-cares during these clock

Table 1: Broadside test set B for $s27$

i	SI_i	A_{i1}	A_{i2}
0	001	1110	1001
1	111	0110	0010
2	101	1001	0010
3	000	1001	0111
4	100	0000	1111
5	010	0010	1110
6	100	1011	1100
7	101	0110	0001
8	110	0100	0011
9	111	0010	1001
10	101	0111	0010
11	000	0001	0010

cycles. The first three vectors of the transparent-scan sequence are thus xxxx11, xxxx10, xxxx10. During the next two clock cycles the circuit operates in functional mode and the primary input vectors $A_{01} = 1110$ and $A_{02} = 1001$ are applied. This requires setting $s_{sel}(3) = 0$ and $s_{sel}(4) = 0$, while $s_{inp}(3)$ and $s_{inp}(4)$ are don't-cares. The corresponding two vectors of the transparent-scan sequence are 11100x, 10010x. A fast capture cycle is used for the second functional clock cycle. Slow capture cycles are used for the other time units (scan clock cycles are thus associated with slow capture cycles). While scanning out the final state of t_0, it is possible to scan in the state SI_1 of t_1. The first clock cycles of the transparent-scan sequence obtained from the broadside test set of Table 1 are shown in Table 2.

Table 2: Transparent-scan sequence

u	$a_0 a_1 a_2 a_3$	s_{sel}	s_{inp}	ρ
0	xxxx	1	1	0
1	xxxx	1	0	0
2	xxxx	1	0	0
3	1110	0	x	0
4	1001	0	x	1
5	xxxx	1	1	0
6	xxxx	1	1	0
7	xxxx	1	1	0
8	0110	0	x	0
9	0010	0	x	1
10	xxxx	1	1	0
11	xxxx	1	0	0
12	xxxx	1	1	0
13	1001	0	x	0
14	0010	0	x	1
15	xxxx	1	0	0
16	xxxx	1	0	0
17	xxxx	1	0	0
18	1001	0	x	0
19	0111	0	x	1
20	xxxx	1	0	0
21	xxxx	1	0	0
22	xxxx	1	1	0
23	0000	0	x	0
24	1111	0	x	1
25	xxxx	1	0	0
26	xxxx	1	1	0
27	xxxx	1	0	0
	...			

The total number of clock cycles required for applying the transparent-scan sequence T is equal to the

number of clock cycles required for applying the broadside test set B from which it is translated. This is also equal to the length of T (the number of vectors included in T). In the case of $s27$, the length of T is 63.

Fault simulation of the transparent-scan sequence T under transition faults is performed by a sequential fault simulation procedure for stuck-at faults, with the following modifications. Simulation of the fault free circuit is done as for stuck-at faults. Simulation of a fault f is identical to fault free simulation in a time unit u where a slow capture cycle is used ($\rho(u) = 0$). To consider a time unit u with a fast capture cycle ($\rho(u) = 1$), suppose that the fault under consideration is the $v \rightarrow v'$ transition fault on line g. If $g = v$ at time unit $u-1$ and $g = v'$ at time unit u, we simulate the fault g stuck-at v at time unit u. If the fault is propagated to an output (an original primary output or a scan chain output), the fault is detected and simulation of f terminates. Otherwise, the next-state obtained is used as the present-state of the faulty circuit at the next time unit. This simulation process uses the fact that a fast capture cycle is always preceded by a slow capture cycle.

For illustration, we consider the $1 \rightarrow 0$ transition fault on primary input a_0 of $s27$ under the test sequence shown in Table 2. The fault is activated for the first time at time unit $u = 14$. We show the states traversed by the fault free and faulty circuits, and the output vectors they produce at time unit 13 to 17 in Table 3. The state vectors are given over the state variables of the circuit, $y_0 y_1 y_2$. The output vectors are given over the original primary output of the circuit, b_0, and the scan output, s_{out}. The fault is detected on the scan output at time unit 17.

Table 3: Fault simulation

u	$a_0 a_1 a_2 a_3$	s_{sel}	s_{inp}	ρ	$y_0 y_1 y_2$	$b_0 s_{out}$
13	1001	0	x	0	101/101	11/11
14	0010	0	x	1	101/101	11/11
15	xxxx	1	0	0	000/100	x0/10
16	xxxx	1	0	0	000/010	x0/x0
17	xxxx	1	0	0	000/001	x0/11

3. Test compaction by vector omission

The length of the transparent-scan sequence T can be reduced by using static test compaction procedures for stuck-at faults in synchronous sequential circuits. We use the vector restoration based static test compaction procedure from [11]. This procedure omits test vectors from the test sequence without reducing the fault coverage. In our case, the set of target faults is the set of transition faults detected by B (and T before compaction).

The restoration based procedure starts by omitting all the test vectors from the test sequence, except for a prefix that synchronizes the fault free circuit. The synchronizing prefix is maintained since synchronization helps in the detection of all the faults. The procedure then restores input vectors into the sequence so as to restore the detection of the target faults originally detected by the sequence. Test vectors that are not restored are omitted from the sequence at the end of the compaction procedure in order to reduce its length.

The restoration based process may omit arbitrary vectors from the test sequence. When considering transition faults, it is necessary to make sure that the original structure of the two-pattern tests embedded in the test sequence is retained after compaction. Specifically, it is important to ensure the existence of consecutive time units $u, u+1$ such that u is a slow capture cycle while $u+1$ is a fast capture cycle, and both u and $u+1$ are functional clock cycles. Such pairs of clock cycles correspond to broadside tests embedded in T. While it is possible to allow consecutive fast capture cycles, this complicates the simulation of transition faults [12]. We therefore prefer to avoid the introduction of consecutive fast capture cycles. We address these issues as follows.

If the test vector at time unit $u+1$ is restored, and if $u+1$ is a fast capture cycle, we also restore the test vector at time unit u at the same time. In the translated test sequence before it is compacted, if $u+1$ is a fast capture cycle, then u is a slow capture cycle and both u and $u+1$ are functional clock cycles. By restoring both u and $u+1$ simultaneously, we ensure that the broadside test embedded at time units $u, u+1$ is maintained in the compacted test sequence.

After test set translation, the transparent-scan sequence T is incompletely specified. We first apply the vector restoration based static test compaction procedure to the incompletely specified sequence T. We then specify the unspecified values of the compacted sequence randomly. Finally, we apply the static test compaction procedure to the fully-specified test sequence.

In the case of $s27$, application of static test compaction to the sequence of Table 2 reduces the length of the sequence from 63 to 49. Application of static test compaction to the fully-specified sequence further reduces its length to 47. The first clock cycles of the compacted sequence obtained for $s27$ are shown in Table 4. It can be seen that broadside tests are still embedded in T after compaction, indicated by $\rho(u) = 0$, $\rho(u+1) = 1$ and $s_{sel}(u) = s_{sel}(u+1) = 0$. Static test compaction mostly removed scan clock cycles that are not necessary.

4. Embedding additional tests

In this section we consider the modification of a compacted fully-specified transparent-scan sequence T so as to create additional opportunities for faults to be detected. Application of static test compaction following the modification of the sequence is expected to further reduce the sequence length. We denote the length of T by L.

We consider a pair of time units $u, u+1$ such that $0 \le u, u+1 < L$ for modification if $\rho(u) = \rho(u+1) = 0$. In addition, if $u+2 < L$ we require that $\rho(u+2) = 0$. In this

Table 4: Compacted and fully-specified sequence

u	$a_0a_1a_2a_3$	s_{sel}	s_{inp}	ρ
0	0011	1	1	0
1	1101	1	0	0
2	0011	1	0	0
3	1110	0	0	0
4	1001	0	0	1
5	1111	1	1	0
6	1000	1	1	0
7	0110	0	1	0
8	0010	0	1	1
9	1011	1	1	0
10	1001	0	0	0
11	0010	0	0	1
12	0100	1	0	0
13	1101	1	0	0
14	1000	1	0	0
15	1001	0	1	0
16	0111	0	1	1
17	1011	1	0	0
18	0010	1	1	0
19	0000	0	0	0
20	1111	0	1	1
21	0001	1	1	0
22	1111	1	0	0
23	0010	0	1	0
24	1110	0	0	1
	...			

case, $u,u+1$ is a subsequence of two consecutive slow capture cycles that is not followed by a fast capture cycle. We consider the possibility of turning $u+1$ into a fast capture cycle by setting $\rho(u+1) = 1$.

To ensure that transition faults continue to be detected only by broadside tests, while changing $\rho(u+1)$ from 0 to 1 we also set $s_{sel}(u) = s_{sel}(u+1) = 0$. Following the modification we check whether all the faults continue to be detected by T. If not, we undo all the changes.

We consider every pair of time units $u,u+1$ that satisfies the conditions above, for $u = 0,1,\cdots,L-2$. This process is illustrated next by considering the transparent-scan sequence of $s27$ shown in Table 4.

Considering $u,u+1 = 0,1$, we find that setting $\rho(1) = 1$ and $s_{sel}(0) = s_{sel}(0) = 0$ results in a test sequence that detects all the faults. We therefore accept the change. Time units $u,u+1 = 1,2$ are not considered since $\rho(1) = 1$. Time units $u,u+1 = 2,3$ are not considered since $\rho(4) = 1$. Similarly, $u,u+1 = 3,4$ and $u,u+1 = 4,5$ are not considered. Considering $u,u+1 = 5,6$, we find that setting $\rho(6) = 1$ and $s_{sel}(5) = s_{sel}(6) = 0$ leaves a fault undetected and we undo the change. In a similar way we find that the test sequence cannot be modified based on $u,u+1 = 12,13$ and $u,u+1 = 13,14$. Considering $u,u+1 = 17,18$, we find that setting $\rho(18) = 1$ and $s_{sel}(17) = s_{sel}(18) = 0$ results in a test sequence that detects all the faults. We therefore accept the change. We find that the sequence cannot be modified based on $u,u+1 = 21,22$, $u,u+1 = 30,31$ and $u,u+1 = 35,36$, but it can be modified based on $u,u+1 = 25,26$ and $u,u+1 = 31,32$. The first clock cycles of the modified sequence are shown in Table 5.

Table 5: Modified transparent-scan sequence

u	$a_0a_1a_2a_3$	s_{sel}	s_{inp}	ρ
0	0011	0	1	0
1	1101	0	0	1
2	0011	1	0	0
3	1110	0	0	0
4	1001	0	0	1
5	1111	1	1	0
6	1000	1	1	0
7	0110	0	1	0
8	0010	0	1	1
9	1011	1	1	0
10	1001	0	0	0
11	0010	0	0	1
12	0100	1	0	0
13	1101	1	0	0
14	1000	1	0	0
15	1001	0	1	0
16	0111	0	1	1
17	1011	0	0	0
18	0010	0	1	1
19	0000	0	0	0
20	1111	0	1	1
21	0001	1	1	0
22	1111	1	0	0
23	0010	0	1	0
24	1110	0	0	1
	...			

Table 6: Modified and compacted sequence

u	$a_0a_1a_2a_3$	s_{sel}	s_{inp}	ρ
0	0011	0	1	0
1	1101	0	0	1
2	0011	1	0	0
3	1110	0	0	0
4	1001	0	0	1
5	1000	1	1	0
6	0110	0	1	0
7	0010	0	1	1
8	1011	1	1	0
9	1001	0	0	0
10	0010	0	0	1
11	0100	1	0	0
12	1101	1	0	0
13	1000	1	0	0
14	1001	0	1	0
15	0111	0	1	1
16	1011	0	0	0
17	0010	0	1	1
18	0000	0	0	0
19	1111	0	1	1
20	0001	1	1	0
21	1111	1	0	0
22	0010	0	1	0
23	1110	0	0	1
24	0101	0	0	0
	...			

After the sequence is modified, we apply the restoration based static test compaction procedure to omit test vectors from T. In the case of $s27$, the modified sequence length is reduced from 47 to 42. We show the first time units of the compacted test sequence in Table 6.

It is also possible to modify the test sequence without requiring that broadside tests would be created. Let us consider a pair of time units $u,u+1$ such that

$\rho(u) = \rho(u+1) = 0$, and either $u+2 = L$ or $\rho(u+2) = 0$. It is possible to modify $\rho(u+1)$ from 0 to 1 without modifying $s_{sel}(u)$ and $s_{sel}(u+1)$. If $s_{sel}(u) = 1$ or $s_{sel}(u+1) = 1$, the resulting test would involve a scan clock cycle at one or both time units.

For the test sequence of $s27$ shown in Table 6, we find that it is possible to change ρ at time units 12 and 21 without reducing the fault coverage. The corresponding tests involve scan operations at both clock cycles.

5. Overall compaction procedure

We apply the procedures described in the previous sections as follows.

We first apply the restoration based static test compaction procedure to the incompletely-specified test sequence T obtained after test set translation. We then specify T randomly and apply the static test compaction procedure again.

We modify the test sequence T to embed additional broadside tests in it by considering every pair of time units $u, u+1$ for $u = 0, 1, \cdots, L-1$. If at least one new test is embedded, we apply the static test compaction procedure to the modified test sequence. Modification and compaction are repeated until the test length is not reduced and the fault coverage is not increased in the last iteration. The fault coverage may increase due to the changes introduced into the test sequence if the test set B does not detect all the faults detectable by broadside tests.

We then modify the test sequence T to embed additional tests, which may not be broadside tests. We found that for most of the time unit pairs $u, u+1$ considered during this modification, $\rho(u+1)$ can be changed from 0 to 1. As a result, only one pass of modification is typically effective and the level of compaction achieved is low. To increase the number of effective passes and thus improve the levels of compaction that can be achieved, we consider a pair of time units $u, u+1$ with probability 1/3, and we skip over $u, u+1$ with probability 2/3 during a pass over the test sequence.

If at least one new test is embedded during a pass over all the time unit pairs, we apply the static test compaction procedure to the modified test sequence. Modification and compaction are repeated until the test length is not reduced and the fault coverage does not increase in the last iteration. The fault coverage may increase due to the introduction of new types of tests into T, as well as due to compaction.

To speed up fault simulation during the modification of a test sequence we use the following techniques. If the sequence is modified at time units $u, u+1$, faults that are detected before time unit u are not affected by the modification. Such faults are not simulated.

We stop the simulation process as soon as a fault that was detected by T before the modification turns out to be undetected after the modification. Any such fault that remains undetected will cause the modification to be undone, and there is no need to simulate additional faults.

For every fault f, we store the number of modifications that failed because f remained undetected. After a modification is performed, we simulate the faults by decreasing order of the number of times they caused a modification to fail. This order is based on the assumption that a fault that already caused a larger number of modifications to fail is more likely to cause another modification to fail.

6. Experimental results

We apply the process described in Section 5 using a broadside test set B. The test set B is obtained by simulating random broadside tests in subsets of $N = 100000$ tests. If the last subset of N tests detects any faults, we apply another subset. We perform fault simulation to drop detected faults and we store in B tests that detect new faults when they are simulated. We follow this by forward-looking reverse order fault simulation [13] of B in order to reduce its size before it is translated.

The results are shown in Table 7. After the circuit name we show the number of transition faults and the fault coverage achieved by B. This fault coverage is maintained (or increased) throughout the compaction and modification process. Under column *init* we show the length of the incompletely-specified sequence T after translation. Under column *compact* we show the length of T after it is compacted, specified, and then compacted again. We also show the length of T as a percentage of its initial length after translation and before compaction. Under column *modify broadside* we show the same information for T after it is modified and compacted, where the modification is done in order to embed additional broadside tests. Under column *modify arbitrary* we show the same information for T after it is modified and compacted, where the modification is done in order to embed arbitrary additional tests. We also show the fault coverage of T after modification and compaction. The fault coverage may increase due to the use of non-broadside tests as well as due to the modification and compaction of T. The fault coverage never decreases due to modification or compaction. We only show the results of embedding arbitrary tests if they are different from the results obtained after embedding of broadside tests. For example, for $s208$, the test length is reduced and the fault coverage increases due to the embedding of arbitrary tests. For $s344$, the test length does not change but the fault coverage increases due to the embedding of arbitrary tests.

From Table 7 it can be seen that application of the restoration based static test compaction procedure to the translated test sequence, and to the translated test sequence after it is fully specified, results in reductions in

test length. Additional reductions are obtained after the sequence is modified so as to introduce additional broadside tests. The introduction of additional types of tests further reduces the test sequence length in some cases, and in many cases increases the fault coverage.

7. Concluding remarks

We showed that a broadside test set for transition faults can be translated into a test sequence under transparent-scan that applies the same broadside tests to the circuit. Considering the transparent-scan sequence, several test compaction techniques were developed. A vector restoration based procedure for static test compaction considering stuck-at faults was extended to delay faults tested by broadside tests under transparent-scan. To improve the effectiveness of static test compaction, the test sequence was modified to include additional broadside tests by modifying slow capture cycles into fast capture cycles while replacing scan clock cycles by functional clock cycles. The transparent-scan sequence was also modified to include other types of tests for delay faults. These modifications are unique to delay faults and were not considered in the context of stuck-at faults earlier. Experimental results for transition faults demonstrated significant reductions in test application times for benchmark circuits.

References

[1] I. Pomeranz and S. M. Reddy, "A New Approach to Test Generation and Test Compaction for Scan Circuits", in Proc. Design Autom. and Test in Europe Conf., March 2003, pp. 1000-1005.

[2] S. P. Morley and R. A. Marlett, "Selected Length Partial Scan: A Method to Reduce Vector Length", in Proc. Intl. Test Conf., 1991, pp. 385-392.

[3] P.-C. Chen, B.-D. Liu and J.-F. Wang, "Overall Consideration of Scan Design and Test Generation", in Proc. Intl. Conf. on Computer-Aided Design, 1992, pp. 9-12.

[4] Y. Higami, S. Kajihara and K. Kinoshita, "Reduced Scan Shift: A New Testing method for Sequential Circuits", in Proc. Intl. Test Conf., 1994, pp. 624-630.

[5] W.-J. Lai, C.-P Kung and C.-S. Lin, "Test Time Reduction in Scan Designed Circuits", in Proc. European Design Autom. Conf., 1993, pp. 489-493.

[6] C. Su and K. Hwang, "A Serial Scan Test Vector Compression Methodology", in Proc. Intl. Test Conf., 1993, pp. 981-988.

[7] I. Pomeranz and S. M. Reddy, "On Improving the Stuck-at Fault Coverage of Functional Test Sequences by Using Limited-Scan Operations", in Proc. Intl. Test Conf., 2001, pp. 211-220.

[8] Y. Cho, I. Pomeranz and S. M. Reddy, "Test Application Time Reduction for Scan Circuits Using Limited Scan Operations", in Proc. Intl. Symp. on Quality Electronic Design, April 2004, pp. 211-216.

[9] J. Grodstein, D. Bhavsar, V. Bettada and R. Davies, "Automatic Generation of Critical-Path Tests for a Partial-Scan Microprocessor", in Proc. Intl. Conf. on Computer Design, Oct. 2003.

[10] J. Savir and S. Patil, "Broad-Side Delay Test", IEEE Trans. on Computer-Aided Design, Aug. 1994, pp. 1057-1064.

[11] I. Pomeranz and S. M. Reddy, "Vector Restoration Based Static Compaction of Test Sequences for Synchronous Sequential Circuits", in Proc. Intl. Conf. on Computer Design, Oct. 1997, pp. 360-365.

[12] I. Pomeranz and S. M. Reddy, "At-Speed Delay Testing of Synchronous Sequential Circuits", in Proc. 29th Design Autom. Conf., June 1992, pp. 177-181.

[13] I. Pomeranz and S. M. Reddy, "Forward-Looking Fault Simulation for Improved Static Compaction", IEEE Trans. on Computer-Aided Design, Oct. 2001, pp. 1262-1265.

Table 7: Experimental results

				compact		modify broadside		modify arbitrary		
circuit	trans	f.c.	init	len	%left	len	%left	len	%left	f.c.
s208	416	77.16	368	305	82.88	276	75.00	258	70.11	77.88
s298	596	81.71	686	454	66.18	340	49.56	-	-	-
s344	688	94.48	678	482	71.09	375	55.31	-	-	95.93
s382	764	78.40	1010	832	82.38	618	61.19	-	-	-
s386	772	79.27	518	475	79.27	373	72.01	360	69.50	82.77
s400	800	77.12	1010	842	83.37	649	64.26	639	63.27	-
s420	840	72.26	1276	947	74.22	901	70.61	823	64.50	74.52
s510	1020	89.90	734	717	97.68	525	71.53	516	70.30	92.45
s526	1052	64.64	1723	1488	86.36	1189	69.01	-	-	-
s641	1280	94.77	1783	920	51.60	674	37.80	651	36.51	96.17
s820	1640	80.73	992	912	91.94	834	84.07	811	81.75	83.72
s953	1906	94.65	4121	1416	34.36	884	21.45	873	21.18	-
s1196	2392	98.91	4798	780	16.26	737	15.36	707	14.74	98.95
s1423	2846	87.63	10182	7063	69.37	5108	50.17	4931	48.43	89.11
b03	768	94.01	1758	1023	58.19	864	49.15	-	-	-
b04	2284	90.02	5982	2991	50.00	2556	42.73	2522	42.16	90.72
b06	332	84.64	284	237	83.45	135	47.54	-	-	-
b09	678	83.48	1018	813	79.86	714	70.14	-	-	-
b10	870	85.06	1290	1028	79.69	740	57.36	-	-	-
b11	1830	85.19	3422	2590	75.69	1891	55.26	-	-	85.63