# A New Approach to Test Generation and Test Compaction for Scan Circuits

Irith Pomeranz[1]
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907

and

Sudhakar M. Reddy[2]
Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242

## Abstract

We propose a new approach to test generation and test compaction for scan circuits that eliminates the distinction between scan operations and application of primary input vectors. Under this approach, the scan-in, scan-select and scan-out lines are treated as conventional primary inputs or primary outputs of the circuit. As a result, limited scan operations, where scan chains are shifted a number of times smaller than their lengths, are incorporated naturally into the test sequences generated by this approach. This leads to very aggressive compaction, resulting in test sequences with the lowest known test application times for benchmark circuits.

## 1. Introduction

Test generation procedures for scan circuits take one of two approaches to the use of scan. For simplicity, we describe them considering full-scan circuits.

Under the first approach, considered in [1]-[5], the present state variables (the outputs of the scan flip-flops) are considered as primary inputs of the circuit, the next state variables (the inputs of the scan flip-flops) are considered as primary outputs of the circuit, and combinational test generation is carried out to detect faults in the combinational logic of the circuit. A test vector $t$ that assigns the vector $t_I$ to the primary inputs and the vector $t_S$ to the present state variables yields the scan-based test $(t_S,t_I)$. The test starts by scanning in $t_s$, then the primary input vector $t_I$ is applied, and the final state reached is scanned out.

Under the second approach, considered in [6]-[9] as well as in commercial tools, the test generation procedure repeatedly selects between two options. The first option is to scan-out the current state and scan-in a new state. The second option is to continue applying primary input vectors without using the scan chain. The choice in [6]-[8] is made so as to minimize the number of clock cycles it will take to detect a target fault. The result is scan-based tests of the form $(SI,T)$ where $SI$ is the scan-in vector and $T$ is a primary input sequence that consists of one or more primary input vectors. A similar test application scheme is used in [10].

In general, we represent a scan-based test generated by the first or second approach as $(SI,T)$ where $T$ can consist of one or more primary input vectors. Under the first approach, $T$ consists of a single primary input vector, and a scan operation occurs before and after every primary input vector is applied. Under the second approach, several primary input vectors may be applied between scan operations.

We observe that the second approach requires a simplified version of a test generation procedure for non-scan sequential circuits in order to generate sequences of primary inputs $T$ of length larger than one. The procedure is simplified by limiting the length of the sequences $T$ and since the initial state of the circuit is controllable and the final state is observable. Although the test generation process is more complicated under the second approach than under the first approach, the second approach has the advantage that it allows the number of scan operations to be reduced by using primary input vectors instead of scan operations to set the circuit state or propagate fault effects. Since every scan operation takes $N_{SV}$ clock cycles (where $N_{SV}$ is the number of state variables in a scan chain) while a primary input vector takes a single clock cycle, the overall result of using longer primary input sequences and fewer scan operations is that the test application time is reduced significantly.

In all the procedures discussed above, a scan-in/out operation observes the values of *all* the flip-flops and assigns new values to *all* the flip-flops. Such a scan operation is referred to as a *complete* scan operation. It is also possible to use *limited* scan operations. For a circuit with $N_{SV}$ state variables included in a single scan chain, the scan chain is shifted by $N_{SH}$ positions under a limited scan operation, where $N_{SH} \leq N_{SV}$. Limited scan operations were used under the first approach in [11]-[15] and under the second approach in [16]. They are useful in producing test sets having reduced test application times since they require fewer scan shift operations than complete scan operations.

Even when limited scan operations are used, there is a clear distinction between scan operations and application of primary input vectors under both approachs. Under the first approach, this is expressed by the fact that scan operations take place before and after every primary input

1. Research supported in part by NSF Grant No. CCR-0098091 and in part by SRC Grant No. 2001-TJ-950.
2. Research supported in part by NSF Grant No. CCR-0097905 and in part by SRC Grant No. 2001-TJ-949.

vector. This is predetermined and cannot be changed. Under the second approach, this is expressed by the fact that explicit choices are made between scan operations and application of primary input vectors. It should be noted that all the existing test generation procedures using the second approach perform only complete scan operations. Thus, they do not take advantage of the flexibility provided by limited scan.

In this work, we propose a new approach to test generation for scan circuits. Under this approach, the distinction between scan operations and application of primary input vectors is eliminated. This allows us to use limited scan operations in the most flexible way and achieve aggressive reductions in test application time.

To describe the proposed approach, we use the following notation. We denote the non-scan circuit by $C$ and the scan circuit by $C_{scan}$. The scan circuit $C_{scan}$ has two extra inputs compared to $C$. The extra inputs are *scan_sel*, which is the select input of the scan multiplexers, and *scan_inp*, which is the input of the (single) scan chain. It also has an extra output, *scan_out*, which is the output of the scan chain.

We note that it is possible to generate tests for $C_{scan}$ using any test generation procedure for non-scan synchronous sequential circuits [17]-[21]. Such a procedure will not distinguish between scan operations and application of primary input vectors, and will thus achieve our goal of eliminating the distinction between them. The test generation effort will be reduced compared to a non-scan circuit since the extra inputs *scan_sel* and *scan_inp* and the extra output *scan_out* provide the state controllability and observability provided by scan. However, we also observe that such a test generation procedure will lack the functional-level knowledge that the circuit contains a scan chain. As a result, it may fail to detect certain faults. We introduce the functional-level knowledge that the circuit has scan into a test generation procedure for non-scan circuits in order to maximize the fault coverage it achieves.

We also observe that it is possible to start from a test set $S$ generated by a procedure that takes the first or second approach (i.e., a procedure that distinguishes between scan operations and application of primary input vectors) and eliminate the distinction between scan operations and application of primary input vectors that exists in $S$. We do this by using explicitly the inputs *scan_sel* and *scan_inp* to define a test sequence $T$. This is useful as discussed next.

Once a test sequence is available under the approach proposed here, it is possible to apply static test compaction procedures designed for non-scan synchronous sequential circuits [22]-[25] in order to reduce the test application time. Here, we note the following. Static test compaction procedures developed for scan circuits distinguish between scan operations and application of primary input vectors, similar to test generation procedures [1]-[9]. Therefore, they suffer from similar limitations. Specifically, these procedures are designed for test sets that use only complete scan operations. When they eliminate a scan operation in order to compact the test set, they eliminate it completely. As a result, they do not have the ability to replace a complete scan operation with a limited one in order to reduce the test application time when the scan operation cannot be eliminated completely. After eliminating the distinction between scan operations and application of primary input vectors, compaction procedures developed for non-scan synchronous sequential circuits can be used for scan circuits. Such procedures have complete flexibility in modifying complete scan operations into limited ones and they achieve significant levels of compaction.

We consider circuits with a single scan chain in this work. However, all the procedures developed can be easily applied to circuits with multiple scan chains. The proposed approach does not impose any special conditions on the tester beyond the requirement to apply limited scan operations, which also exists in [11]-[16].

The paper is organized as follows. In Section 2 we consider the embedding of the functional-level knowledge that the circuit has scan in a test generation procedure for non-scan circuits. In Section 3 we show how a test set generated for a scan circuit can be modified into a test sequence where the distinction between scan operations and application of primary input vectors is eliminated. In Section 4 we consider the effects of using static test compaction procedures for non-scan synchronous sequential circuits on scan circuits. In Section 5 we provide experimental results of test generation and test compaction. In Section 6 we include concluding remarks.

## 2. Test generation procedure

Test generation under the proposed approach, which eliminates the distinction between scan operations and application of primary input vectors, can be done as follows. The circuit for which test generation is carried out is $C_{scan}$. This circuit has two extra primary inputs compared to the original circuit $C$: the scan-in input *scan_inp*, and the scan-select input *scan_sel*. It also has an extra primary output, the scan output *scan_out*. Any test generation procedure for non-scan circuits can be applied to $C_{scan}$ [17]-[21]. The procedure will produce a test sequence where *scan_sel* and *scan_inp* are used as conventional primary inputs, and fault effects may be observed on *scan_out*. An example of such a test sequence is shown in Table 1. This sequence was generated for $s27_{scan}$, which is the scan version of ISCAS-89 benchmark circuit $s27$. The circuit has four primary inputs labeled $a_1, a_2, a_3, a_4$. It has three state variables. It is interesting to note that scan is applied for a single time unit at time unit 5, 7 and 16. In addition, it is applied for two consecutive time units at time units 13, 14 and 18, 19. Thus, all the scan operations are limited scan operations with one and two shifts of the scan chain, and there is never a complete scan operation that takes three shifts of the scan chain.

The disadvantage of using a test generation procedure for a non-scan circuit is that it does not possess the functional-level knowledge that the circuit has a scan

**Table 1: Test sequence for $s27_{scan}$**

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | *scan_sel* | *scan_inp* |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 0 |
| 8 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 1 |
| 12 | 0 | 0 | 0 | 1 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 1 | 0 |
| 14 | 0 | 0 | 0 | 0 | 1 | 1 |
| 15 | 0 | 0 | 0 | 1 | 0 | 0 |
| 16 | 1 | 0 | 0 | 0 | 1 | 0 |
| 17 | 0 | 0 | 0 | 1 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 1 | 1 |
| 19 | 0 | 0 | 0 | 0 | 1 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 1 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 1 | 0 | 0 | 0 |
| 23 | 1 | 0 | 0 | 1 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 |

chain. To demonstrate the importance of this knowledge, consider the case where the circuit has $N_{SV} = 100$ scan state variables. Consider a fault whose effect can only be propagated to the first flip-flop in the scan chain. Thus, it is not possible to propagate the fault effect to a primary output (other than *scan_out*) or directly to any other flip-flop. In this case, it will take 100 clock cycles where *scan_sel* $= 1$ to scan-out the fault effect and observe it on the scan output *scan_out*. A typical gate-level test generation procedure is not likely to find such a test sequence of length ≥ 100. However, it is possible to modify the test generation procedure to identify cases where fault effects reach the flip-flops. In these cases it is possible to use sequences of input vectors where *scan_sel* $= 1$ in order to bring the fault effects to the *scan_out* output. We introduce this modification into a test generation procedure for non-scan circuits as follows.

The test generation procedure we use for non-scan circuits constructs a test sequence $T$ by concatenating test subsequences for yet-undetected target faults. Thus, $T$ is initially empty. At an arbitrary stage of test generation, a target fault $f$ is selected and an attempt is made to generate a test subsequence $\hat{T}$ such that $T\hat{T}$ detects $f$. If $\hat{T}$ is found, it is concatenated to $T$ to obtain $T = T\hat{T}$. The subsequence $\hat{T}$ is generated forward in time starting from the final state reached under $T$.

In the modified procedure, when a target fault $f$ is selected, we first attempt to generate a test subsequence $\hat{T}$ for $f$ using the original process. If $\hat{T}$ cannot be found, we check whether a fault effect of $f$ was propagated to a flip-flop at any point during the attempt to generate $\hat{T}$. If the fault effect was propagated to flip-flop $i$ by a subsequence $T'$ obtained during the test generation process, we define the following test subsequence $\hat{T}$ for $f$. We note that it is possible to propagate the fault effect from flip-flop $i$ to the scan output by applying $N_{SV} - i$ primary input vectors where *scan_sel* $= 1$. The first vector will propagate the fault effect from flip-flop $i$ to flip-flop $i+1$, the second vector will propagate the fault effect from flip-flop $i+1$ to flip-flop $i+2$, and so on, until the last vector will cause the fault effect to appear on *scan_out*. We define a test subsequence $T''$ that consists of $N_{SV} - i$ primary input vectors where *scan_sel* $= 1$. We fill the remaining primary input values under $T''$ randomly. We then define $\hat{T} = T'T''$. By concatenating $\hat{T}$ to $T$, we guarantee that $f$ would be detected.

For example, we consider fault $f_{132}$ of $s298_{scan}$. The fault cannot be detected by the original test generation process. However, its effects are propagated to a flip-flop by the subsequence $T'$ = 00011 00100 10000 obtained during the test generation process. As in Table 1, we show the values of *scan_sel* and *scan_inp* in this order at the end of every primary input vector. The circuit has 14 flip-flops, and the fault effect reaches flip-flop 5. Therefore, to propagate the fault effect to *scan_out*, it is necessary to apply 9 primary input vectors where *scan_sel* $= 1$. The subsequence $T''$ = 00111 11011 00111 00111 11110 00111 10111 00011 00111 is used for this purpose. Note that *scan_sel* $= 1$ in every vector, while the other input values are set randomly. We obtain the test sequence $T'T''$ for $f_{132}$.

The test generation procedure we use for non-scan circuits processes time units in the forward direction only. Therefore, we use the functional-level knowledge of the existence of a scan chain only when propagation of fault effects fails. In a test generation procedure that performs both forward and backward time processing such as [20], it is also possible to use the functional-knowledge of scan when a state cannot be justified. If a fault can be activated (and propagated) starting from a state $s$ but $s$ cannot be justified, a sequence of length $N_{SV}$ with *scan_sel* $= 1$ in every vector and *scan_inp* set appropriately can be used to bring the circuit to state $s$. For example, let $s$ = 00111. Let $\alpha$ denote an arbitrary vector on the original primary inputs of the circuit, and let the values of *scan_sel* and *scan_inp* be the last two values in every primary input vector. With $T''$ = $\alpha$11 $\alpha$11 $\alpha$11 $\alpha$10 $\alpha$10, the circuit can be brought to state $s$ in five clock cycles. Note that we reversed the state $s$ in order to bring three 1s to the last three state variables in the scan chain and two 0s to the first two state variables in the scan chain.

When we perform test generation for $C_{scan}$, we consider faults in the logic added in order to implement a scan chain. Thus, the number of faults we target is higher than the number of faults typically targeted by the first or second approach.

## 3. Test set translation

Instead of using the test generation procedure of Section 2, it is also possible to translate a test set $S$ generated under the first or second approach into a test sequence $T$ under the proposed approach.

We demonstrate the translation of a test set $S$ into a test sequence $T$ by considering the test set $S$ for $s27_{scan}$ shown in Table 2. The test set $S$ consists of four tests, ($SI_1$,$T_1$), ($SI_2$,$T_2$), ($SI_3$,$T_3$) and ($SI_4$,$T_4$). The sequences $T_i$ are defined over the original primary inputs of $s27$. Our goal is to combine the tests into a single test sequence $T$ where scan operations are implemented by setting $scan\_sel = 1$ and assigning the appropriate values to $scan\_inp$.

**Table 2: Test set $S$ for $s27_{scan}$**

| $i$ | $SI_i$ | $T_i$ |
|---|---|---|
| 1 | 011 | 0000 |
| 2 | 011 | 1101 |
| 3 | 000 | 1010 |
| 4 | 110 | 0100 0111 1001 |

In the example of Table 2, we start with three vectors to bring the circuit to state $SI_1 = 011$. These vectors contain arbitrary values on the original primary inputs $a_1a_2a_3a_4$ of the circuit. We mark an arbitrary value by an x. The *scan_sel* input must be set to 1 in these vectors, and *scan_inp* must be such that the state 011 will result. This dictates that the first three vectors must be xxxx11 xxxx11 xxxx10. We then apply the primary input sequence $T_1$ = 0000 while holding $scan\_sel = 0$ and *scan_inp* = x. Next, we apply the three vectors xxxx11 xxxx11 xxxx10 in order to scan-out the current state and scan-in the state $SI_2 = 011$. We then apply the primary input sequence $T_2$ = 1101. Continuing in the same way, we obtain the test sequence shown in Table 3.

**Table 3: Test sequence based on $S$ for $s27_{scan}$**

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | *scan_sel* | *scan_inp* |
|---|---|---|---|---|---|---|
| 0 | x | x | x | x | 1 | 1 |
| 1 | x | x | x | x | 1 | 1 |
| 2 | x | x | x | x | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | x |
| 4 | x | x | x | x | 1 | 1 |
| 5 | x | x | x | x | 1 | 1 |
| 6 | x | x | x | x | 1 | 0 |
| 7 | 1 | 1 | 0 | 1 | 0 | x |
| 8 | x | x | x | x | 1 | 0 |
| 9 | x | x | x | x | 1 | 0 |
| 10 | x | x | x | x | 1 | 0 |
| 11 | 1 | 0 | 1 | 0 | 0 | x |
| 12 | x | x | x | x | 1 | 0 |
| 13 | x | x | x | x | 1 | 1 |
| 14 | x | x | x | x | 1 | 1 |
| 15 | 0 | 1 | 0 | 0 | 0 | x |
| 16 | 0 | 1 | 1 | 1 | 0 | x |
| 17 | 1 | 0 | 0 | 1 | 0 | x |
| 18 | x | x | x | x | 1 | x |
| 19 | x | x | x | x | 1 | x |
| 20 | x | x | x | x | 1 | x |

We randomly specify all the unspecified values contained in $T$. The resulting test sequence is guaranteed to detect all the faults detected by $S$.

## 4. Static test compaction

Several static test compaction procedures have been developed for non-scan synchronous sequential circuits, which accept a single test sequence [22]-[25]. Any one of these procedures can be applied to a test sequence produced for the circuit $C_{scan}$ under the proposed approach.

For illustration, we applied the procedure based on vector restoration from [23] followed by the procedure based on vector omission from [22] to $s27_{scan}$. We used the test sequence shown in Table 1. The resulting sequence is shown in Table 4. It can be seen that the compaction procedures omitted one primary input vector for which $scan\_sel = 1$, and they created two subsequences of consecutive vectors where $scan\_sel = 1$, one of length two and one of length four.

**Table 4: Compacted test sequence for $s27_{scan}$**

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | *scan_sel* | *scan_inp* |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 1 |
| 9 | 0 | 0 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 0 | 0 | 1 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 1 |
| 12 | 0 | 0 | 0 | 0 | 1 | 0 |
| 13 | 0 | 1 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 1 | 0 | 0 | 0 |
| 15 | 1 | 0 | 0 | 1 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 |

## 5. Experimental results

In this section, we report the results of test generation and test compaction for ISCAS-89 and ITC-99 benchmark circuits into which we inserted scan chains. The order of the flip-flops in the scan chains is identical to their order in the circuit description.

We first applied the test generation procedure described in Section 2. The results are shown in Tables 5 and 6. In Table 5, after the circuit name, we show the number of primary inputs (including the scan-select and scan-in inputs), the number of state variables, and the number of faults (including faults in the multiplexers we added to implement scan chains). Under column *detected* we show the total number of detected faults, the fault coverage, and the number of faults detected by using functional-level knowledge of scan (i.e., using the ability to detect a fault once it reaches a flip-flop).

In Table 6, after the circuit name, we show the total number of primary input vectors and the number of primary input vectors with $scan\_sel = 1$ for the following test sequences. (1) Under column *test len* we show the information for the test sequence generated by the procedure described in Section 2. We denote this sequence by $T$. (2) Under column *restor len* we show the information for the test sequence obtained by applying to $T$ the vector restoration based compaction procedure from [23]. We denote this sequence by $T_{restor}$. (3) Under column *omit len* we show the information for the test sequence obtained by applying to $T_{restor}$ the vector omission based

**Table 5: Fault coverage after test generation**

| circ | inp | stvr | faults | detected total | detected fcov | detected funct |
|---|---|---|---|---|---|---|
| s208 | 13 | 8 | 267 | 266 | 99.63 | 0 |
| s298 | 5 | 14 | 398 | 398 | 100.00 | 3 |
| s344 | 11 | 15 | 452 | 452 | 100.00 | 0 |
| s382 | 5 | 21 | 541 | 535 | 98.89 | 6 |
| s386 | 9 | 6 | 424 | 424 | 100.00 | 0 |
| s400 | 5 | 21 | 566 | 555 | 98.06 | 6 |
| s420 | 21 | 16 | 530 | 523 | 98.68 | 3 |
| s444 | 5 | 21 | 616 | 598 | 97.08 | 12 |
| s510 | 21 | 6 | 604 | 603 | 99.83 | 0 |
| s526 | 5 | 21 | 687 | 673 | 97.96 | 20 |
| s641 | 37 | 19 | 623 | 619 | 99.36 | 0 |
| s820 | 20 | 5 | 884 | 868 | 98.19 | 0 |
| s953 | 18 | 29 | 1299 | 1298 | 99.92 | 30 |
| s1196 | 16 | 18 | 1374 | 1368 | 99.56 | 5 |
| s1423 | 19 | 74 | 1987 | 1947 | 97.99 | 34 |
| s1488 | 10 | 6 | 1526 | 1525 | 99.93 | 0 |
| s5378 | 37 | 179 | 5797 | 5381 | 92.82 | 42 |
| s35932 | 37 | 1728 | 49466 | 42847 | 86.62 | 3 |
| b01 | 5 | 5 | 169 | 169 | 100.00 | 0 |
| b02 | 4 | 4 | 96 | 96 | 100.00 | 0 |
| b03 | 7 | 30 | 636 | 633 | 99.53 | 35 |
| b04 | 14 | 66 | 1746 | 1743 | 99.83 | 28 |
| b06 | 5 | 9 | 268 | 268 | 100.00 | 0 |
| b09 | 4 | 28 | 592 | 587 | 99.16 | 35 |
| b10 | 14 | 17 | 618 | 617 | 99.84 | 6 |
| b11 | 10 | 30 | 1273 | 1254 | 98.51 | 22 |

**Table 6: Test length after test generation and compaction**

| | test len | | restor len | | omit len | | ext | [26] |
|---|---|---|---|---|---|---|---|---|
| circ | total | scan | total | scan | total | scan | det | cyc |
| s208 | 194 | 128 | 155 | 105 | 140 | 94 | | NA |
| s298 | 215 | 90 | 177 | 63 | 161 | 55 | | 218 |
| s344 | 161 | 89 | 105 | 56 | 85 | 48 | | 98 |
| s382 | 811 | 149 | 551 | 118 | 378 | 89 | +3 | 619 |
| s386 | 324 | 157 | 247 | 121 | 216 | 108 | | NA |
| s400 | 766 | 154 | 561 | 119 | 396 | 102 | +2 | 587 |
| s420 | 1353 | 1238 | 550 | 479 | 408 | 363 | | NA |
| s444 | 750 | 286 | 480 | 185 | 450 | 175 | +2 | NA |
| s510 | 278 | 159 | 237 | 128 | 210 | 123 | | NA |
| s526 | 1727 | 703 | 969 | 414 | 726 | 316 | +2 | 1091 |
| s641 | 605 | 451 | 255 | 179 | 239 | 173 | | 302 |
| s820 | 550 | 283 | 443 | 229 | 347 | 183 | +4 | 367 |
| s953 | 1029 | 826 | 448 | 289 | 329 | 210 | | NA |
| s1196 | 928 | 613 | 295 | 179 | 262 | 155 | | NA |
| s1423 | 3148 | 2360 | 1229 | 1011 | 1127 | 953 | +6 | 1816 |
| s1488 | 548 | 280 | 470 | 235 | 416 | 211 | | 416 |
| s5378 | 5381 | 4594 | 2858 | 2601 | 2721 | 2487 | +57 | 18585 |
| s35932 | 634 | 518 | 634 | 518 | 634 | 518 | | 3561 |
| total | | | | | 7230 | | | 27660 |
| b01 | 192 | 79 | 123 | 49 | 89 | 37 | | 61 |
| b02 | 110 | 37 | 73 | 24 | 52 | 17 | | 35 |
| b03 | 1311 | 1152 | 405 | 336 | 347 | 288 | | 588 |
| b04 | 1770 | 1465 | 860 | 671 | 715 | 606 | | 1066 |
| b06 | 140 | 41 | 110 | 34 | 72 | 28 | | 64 |
| b09 | 2026 | 1842 | 789 | 699 | 716 | 635 | | 573 |
| b10 | 959 | 741 | 378 | 272 | 330 | 252 | | 427 |
| b11 | 1797 | 1337 | 1047 | 758 | 789 | 584 | +1 | 986 |
| total | | | | | 3110 | | | 3800 |

compaction procedure from [22].

In some cases, compaction increases the number of detected faults [22]. When this happens, we show the number of extra faults detected after compaction in column *ext det*.

We point out that the test sequence length in our case is equal to the number of clock cycles required to apply the test sequence, since scan operations are represented explicitly in the test sequence. For comparison, we show in the last column of Table 6 the number of clock cycles required to apply the test sets generated for the scan circuit in [26]. This is the best known compaction procedure of its type. Its test application times are lower than those reported in [6]-[9].

In the rows labeled *total* in Table 6 we show total test lengths considering only circuits for which the results from [26] are available.

From Table 5 it can be seen that 100% or close to 100% fault coverage is achieved by the test generation procedure from Section 2. The test generation procedure is not complete, i.e., it is not able to prove that a fault is undetectable. However, it serves to demonstrate that test generation that does not distinguish between scan operations and application of primary input vectors can be carried out effectively on circuits with the scan chain embedded in them.

From Table 6 it can be seen that test compaction procedures for non-scan circuits are able to reduce the test length significantly. The test generation procedure of Section 2 does not use any compaction heuristics. Therefore, the test sequences it generates are relatively long. However, after compaction, we improve on the best known test application times for the benchmark circuits considered under scan shown in the last column of Table 6.

Next, we considered test sets generated by the compaction procedure from [26]. This procedure takes the second approach to test generation for scan circuits. Thus, it distinguishes between scan operations and application of primary input vectors. Using the procedure demonstrated in Section 3, we translated each test set $S$ into a test sequence $T$ under the proposed approach. We then compacted this test sequence using the restoration based procedure from [23] followed by the omission based procedure from [22]. The results are shown in Table 7. We show the test length (the number of primary input vectors) and the number of primary input vectors with $scan\_sel = 1$ for the sequence $T$, for the sequence obtained from $T$ after applying the restoration based compaction procedure, and for the sequence obtained after applying the omission based compaction procedure. For ease of reference, we repeat in the last column the number of clock cycles required for application of the test sets from [26]. In the row labeled *total* we show the total test length over the circuits considered.

From Tables 6 and 7, significant levels of compaction can be achieved by allowing limited scan operations under the proposed approach compared to earlier approaches. This can be seen by comparing the test length under column *omit len* subcolumn *total* of Tables 6 and 7 to the number of cycles from [26]. The results in Table 7 demonstrate that even if the conventional test generation procedures for scan designs are used, test compaction using the approach presented here can significantly reduce

**Table 7: Results for translated test sets**

| | test len | | restor len | | omit len | | [26] |
|---|---|---|---|---|---|---|---|
| circ | total | scan | total | scan | total | scan | cyc |
| s298 | 218 | 140 | 190 | 112 | 172 | 101 | 218 |
| s344 | 98 | 60 | 65 | 28 | 65 | 28 | 98 |
| s382 | 619 | 231 | 534 | 147 | 483 | 125 | 619 |
| s400 | 587 | 231 | 455 | 173 | 364 | 148 | 587 |
| s526 | 1091 | 546 | 870 | 446 | 798 | 387 | 1091 |
| s641 | 302 | 209 | 240 | 161 | 190 | 137 | 302 |
| s820 | 367 | 90 | 350 | 85 | 327 | 78 | 367 |
| s1423 | 1816 | 888 | 1402 | 800 | 1318 | 775 | 1816 |
| s1488 | 416 | 120 | 385 | 105 | 359 | 97 | 416 |
| s5378 | 18585 | 17900 | 11959 | 11832 | 11626 | 11501 | 18585 |
| total | | | | | 15702 | | 24099 |
| b01 | 61 | 10 | 56 | 9 | 56 | 9 | 61 |
| b02 | 35 | 12 | 34 | 11 | 33 | 10 | 35 |
| b03 | 588 | 480 | 421 | 345 | 366 | 307 | 588 |
| b04 | 1066 | 924 | 708 | 570 | 671 | 540 | 1066 |
| b06 | 64 | 36 | 62 | 34 | 60 | 33 | 64 |
| b09 | 573 | 364 | 438 | 242 | 405 | 211 | 573 |
| b10 | 427 | 306 | 346 | 226 | 323 | 204 | 427 |
| b11 | 986 | 480 | 681 | 354 | 662 | 339 | 986 |
| total | | | | | 2576 | | 3800 |

test application times beyond the best known approaches to compact tests for scan designs.

## 6. Concluding remarks

We proposed an approach to test generation and test compaction for scan circuits under which the distinction between scan operations and application of primary input vectors is eliminated. This view is achieved by considering the scan-in, scan-select and scan-out lines as conventional primary inputs or primary outputs of the circuit. This approach has several implications. (1) It allows us to generate test sequences utilizing limited scan operations without explicitly making decisions regarding the number of shifts under each scan operation. (2) It allows us to keep track of circuit states and outputs during scan operations without employing a separate procedure for this purpose and without complicating the test generation process. Thus, this approach provides flexibility in test generation and test compaction that does not exist under approaches that keep scan operations and application of primary input vectors distinct. We considered two procedures for generating test sequences under this approach. Under the first procedure, test generation was carried out using a test generation procedure for non-scan circuits. The procedure was enhanced by functional-level knowledge that the circuit has scan. Under the second procedure, a test set that distinguishes between scan operations and application of primary input vectors was translated into a test sequence under the proposed approach. We showed that test compaction procedures for non-scan circuits can be used to achieve significant levels of compaction under this approach.

## References


[1] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", IEEE TC, March 1981, pp. 215-222.

[2] H. Fujiwara and T. Shimono, "On the Acceleration of Test Generation Algorithms", IEEE TC, Dec. 1983, pp. 1137-1144.

[3] M. Schulz et al., "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System", IEEE TCAD, Jan. 1988, pp. 126-137.

[4] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits", IEEE TCAD, Dec. 1995, pp. 1496-1504.

[5] I. Hamazaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits", in Proc. ICCAD-98, pp. 283-289.

[6] D. K. Pradhan and J. Saxena, "A Design for Testability Scheme to Reduce Test Application Time in Full Scan", in Proc. VTS-92, pp. 55-60.

[7] S. Y. Lee and K. K. Saluja, "An Algorithm to Reduce Test Application Time in Full Scan Designs", in Proc. 1992 ICCAD-92, pp. 17-20.

[8] S. Y. Lee and K. K. Saluja, "Test Application Time Reduction for Sequential Circuits with Scan", IEEE TCAD, Sept. 1995, pp. 1128-1140.

[9] I. Pomeranz and S. M. Reddy, "Simulation Based Test Generation for Scan Designs", in Proc. ICCAD-00, pp. 544-549.

[10] H. Fujiwara and A. Yamamoto, "Parity-Scan Design to Reduce the Cost of Test Application", IEEE TCAD, Oct. 1993, pp. 1604-1611.

[11] S. P. Morley and R. A. Martlett, "Selected Length Partial Scan: A Method to Reduce Vector Length", in Proc. ITC-91, pp. 385-392.

[12] P.-C. Chen, B.-D. Liu and J.-F. Wang, "Overall Consideration of Scan Design and Test Generation", in Proc. ICCAD-92, pp. 9-12.

[13] Y. Higami, S. Kajihara and K. Kinoshita, "Reduced Scan Shift: A New Testing method for Sequential Circuits", in Proc. ITC-94, pp. 624-630.

[14] W.-J. Lai, C.-P Kung and C.-S. Lin, "Test Time Reduction in Scan Designed Circuits", in Proc. EDAC-93, pp. 489-493.

[15] C. Su and K. Hwang, "A Serial Scan Test Vector Compression Methodology", in Proc. ITC-93, pp. 981-988.

[16] I. Pomeranz and S. M. Reddy, "On Improving the Stuck-at Fault Coverage of Functional Test Sequences by Using Limited-Scan Operations", in Proc. ITC-01, pp. 211-220.

[17] R. Marlett, "An Effective Test Generation System for Sequential Circuits", in Proc. DAC-86, pp. 250-256.

[18] W-T. Cheng and T. J. Chakraborty, "Gentest: An Automatic Test Generation System for Sequential Circuits", IEEE Computer, April 1989, pp. 43-49.

[19] T. P. Kelsey and K. K. Saluja, "Fast Test Generation for Sequential Circuits", in Proc. ICCAD-89, pp. 354-357.

[20] T. Niermann and J. H. Patel, "HITEC: A Test Generation Package for Sequential Circuits", in Proc. EDAC-91, pp. 214-218.

[21] X. Lin, I. Pomeranz and S. M. Reddy, "Techniques for Improving the Efficiency of Sequential Circuit Test Generation", in Proc. ICCAD-99, pp. 147-151.

[22] I. Pomeranz and S. M. Reddy, "On Static Compaction of Test Sequences for Synchronous Sequential Circuits", in Proc. DAC-96, pp. 215-220.

[23] I. Pomeranz and S. M. Reddy, "Vector Restoration Based Static Compaction of Test Sequences for Synchronous Sequential Circuits", in Proc. ICCD-97, pp. 360-365.

[24] S. K. Bommu, S. T. Chakradhar and K. B. Doreswamy, "Static Compaction Using Overlapped Restoration and Segment Pruning", in Proc. ICCAD-98, pp. 140-146.

[25] F. Corno, P. Prinetto, M. Rebaudengo and M. Sonza Reorda, "New Static Compaction Techniques of Test Sequences for Sequential Circuits", in Proc. ED&TC-97, pp. 37-43.

[26] I. Pomeranz and S. M. Reddy, "Test Compaction for At-Speed Testing of Scan Circuits Based on Non-Scan Test Sequences and Removal of Transfer Sequences", IEEE TCAD, June 2002, pp. 706-714.