

# Reducing Test Data Volume Using Random-Testable and Periodic-Testable Scan Chains in Circuits with Multiple Scan Chains

Irith Pomeranz<sup>+</sup>  
School of Electrical & Computer Eng.  
Purdue University  
W. Lafayette, IN 47907

## Abstract

For a scan design with multiple scan chains, we say that a scan chain is random-testable if it is possible to achieve complete fault coverage for the circuit (i.e., detect all the detectable circuit faults) when the scan chain is driven from a source of pseudo-random values for the complete test application process. Similarly, a periodic-testable scan chain can be driven from a source that produces a periodic sequence for the complete test application process. Both pseudo-random and periodic sources are simple to implement on-chip. By identifying scan chains that can be driven from pseudo-random and periodic sources and driving them from the appropriate on-chip sources, we reduce the number of scan chains that need to be driven from an external tester. In this way, we reduce the number of scan inputs that a tester needs to control and the amount of test data that the external tester needs to store and apply to the circuit. Existing test data compression techniques can be used to further reduce the test data volume.

## 1. Introduction

The use of multiple scan chains for a scan design helps reduce the test application time compared to the case where a single scan chain is used. To reduce the test data volume as well, several techniques have been proposed to make the number of scan elements appear to the external tester to be smaller than it actually is. The methods of [1]-[9] drive  $N_{CH}$  internal scan chains of the design through a smaller number  $N_{EXT}$  of external inputs. To the external tester, the design appears to have a smaller number of scan chains, and the test data volume required to drive them is reduced. Internally, the  $N_{EXT}$ -bit vectors applied to the external inputs are expanded into  $N_{CH}$ -bit vectors driving the  $N_{CH}$  internal scan chains. The method of [10] drives the scan chain of the design from a shorter virtual scan chain. To the external tester, the design appears to have a shorter scan chain, and the test data volume required to drive it is reduced. Internally, the values loaded into the shorter virtual scan chain are expanded into vectors that drive the full-length scan chain

of the design. Other methods of expanding a small amount of deterministic test data into a complete test set (one that detects all the detectable circuit faults) were described in [11]-[16]. In [17], test data compression is achieved by using the scan chains of one circuit to expand encoded tests for another circuit.

A different approach to the reduction of test data volume is to use a combination of deterministic and pseudo-random values [9], [18]-[21]. Pseudo-random values can be produced by an *LFSR*. In [20], the scan chains of a design are partitioned into two subsets in a round-robin fashion. For a given partition, one set of scan chains is driven by pseudo-random patterns and the other set of scan chains is driven with deterministic values. The partitions are changed to allow every scan chain to be driven from both types of sources.

Further reductions in test data volume as well as the number of scan inputs that an external tester needs to control can be obtained by classifying the scan chains *permanently* into several categories and driving each one from an appropriate source of test data. The sources do not have to be limited to pseudo-random or deterministic, as we show below. In this work, we distinguish between three classes of scan chains defined as follows.

**Definition 1:** We say that a scan chain  $ch$  is *random-testable* if it is possible to achieve complete fault coverage for the circuit (i.e., detect all the detectable circuit faults) when  $ch$  is driven from a source of pseudo-random values for the complete test application process (which consists of a limited number of clock cycles).

**Definition 2:** We say that a scan chain  $ch$  is *periodic-testable* if it is possible to achieve complete fault coverage for the circuit when  $ch$  is driven from a source that produces a periodic sequence for the complete test application process (which consists of a limited number of clock cycles).

Both random-testable scan chains and periodic-testable scan chains can be driven from simple on-chip sources. Specifically, an *LFSR* can be used for random-testable scan chains, and finite-state machines producing periodic sequences can be used for periodic-testable scan chains. Thus, an external tester does not need to provide test data for such scan chains. We limit the number of

<sup>+</sup> Research supported in part by NSF Grant No. CCR-0098091 and in part by SRC Grant No. 2001-TJ-950.

clock cycles for which a random-testable or periodic-testable scan chain is driven from its source in order to control the test application time, as discussed later.

We point out that other types of scan chains with simple on-chip sources, in addition to random-testable and periodic-testable scan chains, may also exist. We only consider these two types in this work.

**Definition 3:** A *deterministic-testable* scan chain  $ch$  is one which is not random-testable and not periodic-testable.

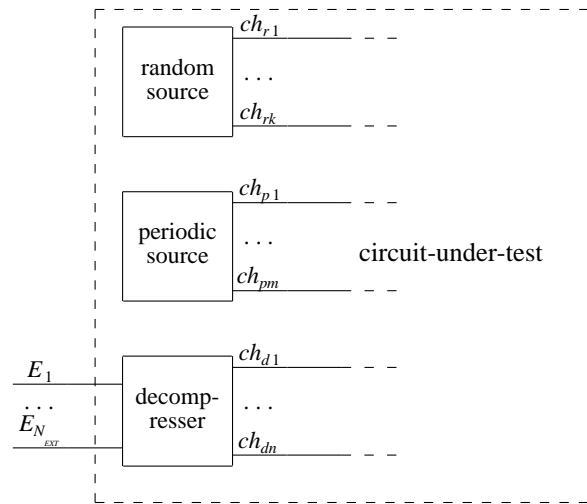
To achieve complete fault coverage, a deterministic-testable scan chain is driven with deterministic values supplied by an external tester. Thus, the external tester needs to provide test data only for deterministic-testable scan chains. The smaller their number, the smaller the amount of test data required. After identifying deterministic-testable scan chains, further compression of test data can be obtained by using any one of the techniques presented in [1]-[21].

To use random-testable and periodic-testable scan chains for reducing the test data volume, it is necessary to identify sets of scan chains that are random-testable or periodic-testable *simultaneously*, i.e., all the scan chains that are not deterministic-testable should be driven from their appropriate sources simultaneously. The following definition addresses this issue.

**Definition 4:** Scan chains  $ch_{r1}, ch_{r2}, \dots, ch_{rk}$  are random-testable and scan chains  $ch_{p1}, ch_{p2}, \dots, ch_{pm}$  are periodic-testable *simultaneously* if it is possible to achieve complete fault coverage for the circuit when the scan chains  $ch_{r1}, ch_{r2}, \dots, ch_{rk}$  are driven from sources of pseudo-random values and the scan chains  $ch_{p1}, ch_{p2}, \dots, ch_{pm}$  are driven from sources that produce periodic sequences for a limited number of clock cycles.

In finding scan chains that are random-testable or periodic-testable simultaneously, it is possible to limit the number of clock cycles for which the chains are driven from their test data sources such that the test application time would be the same as that of a compact deterministic test set, which is applied in full through an external tester. However, this restriction on test application time does not allow us to take full advantage of random-testable or periodic-testable scan chains. We use a parameter  $M$  to denote the increase in test application time that can be tolerated in order to maximize the fault coverage achievable by random and periodic values. Even when  $M > 1$ , we ensure that the input test data volume does not increase for the deterministic-testable scan chains by applying to them the values included in a compact deterministic test set repeatedly. Thus, the tester needs to store only one copy of a deterministic component which is applied repeatedly while using different values from the random and periodic on-chip sources.

We conduct experiments on benchmark circuits and find that there are non-trivial numbers of random-testable and periodic-testable scan chains in these circuits when multiple scan chains are used for them. A block diagram of the proposed method is shown in Figure 1. In Figure 1 we show a circuit with  $k$  random-testable scan chains driven from a source of pseudo-random patterns,  $m$  periodic-testable scan chains driven from a source that produces periodic sequences, and  $n$  deterministic-testable scan chains. The deterministic-testable scan chains can be driven directly from external inputs, or through a *decompressor*. The latter option is shown in Figure 1. Here, the  $n$  deterministic-testable scan chains are driven from  $N_{EXT} < n$  external inputs through a decompressor. The decompressor translates an  $N_{EXT}$ -bit vector applied from an external source into an  $n$ -bit vector which is then shifted in through the deterministic-testable scan chains.



**Figure 1: Circuit configuration**

We describe a procedure for finding scan chains that are random-testable or periodic-testable simultaneously in Section 2. In Section 3 we present experimental results to demonstrate the numbers of deterministic-testable scan chains in benchmark circuits. In Section 4 we demonstrate the use of the proposed definitions together with other compression techniques. Section 5 concludes the paper.

## 2. Identifying random-testable and periodic-testable scan chains

We identify scan chains that are random-testable or periodic-testable simultaneously by using a test set  $T$  based on a compact deterministic test set  $T_0$  for the circuit. We first discuss the partitioning of flip-flops into scan chains and the construction of  $T$ . We then describe the classification of scan chains. We illustrate the procedure by considering ISCAS-89 benchmark circuit *s27*.

## 2.1. Preliminaries

When we consider benchmark circuits, we assume that all the inputs of the combinational logic of the circuit are driven from scan chains. Thus, we do not distinguish between primary inputs and pseudo-inputs. We assume that the order of the flip-flops in the scan chains is identical to the order by which the inputs they drive appear in the circuit description. To partition  $N_I$  inputs into  $N_{CH}$  scan chains, we place approximately equal numbers of consecutive inputs in consecutive scan chains. Thus, to partition inputs  $x_0, x_1, \dots, x_7$  into three scan chains, we place  $x_0, x_1$  in the first scan chain,  $x_2, x_3, x_4$  in the second scan chain, and  $x_5, x_6, x_7$  in the third scan chain. We denote the  $i$ th scan chain by  $ch_i$  and the number of flip-flops included in  $ch_i$  by  $L_i$ . We also use  $ch_i$  to denote the set of inputs driven by  $ch_i$ ,  $ch_i = \{x_{i1}, x_{i1+1}, \dots, x_{i1+L_i-1}\}$ .

For illustration, we consider *s27*, which has seven inputs. We assume that the inputs are divided into four scan chains,  $ch_0 = \{x_0\}$ ,  $ch_1 = \{x_1, x_2\}$ ,  $ch_2 = \{x_3, x_4\}$  and  $ch_3 = \{x_5, x_6\}$ . A compact deterministic test set  $T_0$  for *s27* is shown in Table 1. We denote by  $F_0$  the set of faults detected by  $T_0$ . The set  $F_0$  is the set of target faults we use.

**Table 1: Test set  $T_0$  for *s27***

$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
0	0	0	0	0	1	1
1	0	0	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	1	0	1	0	1	1
1	0	1	0	0	0	0

To identify random-testable and periodic-testable scan chains, we define a test set  $T$  that consists of  $M$  copies of  $T_0$ , where  $M$  is a constant that determines the test application time. With  $M = 1$ , we apply to the circuit a test set of the same size as  $T_0$ , except that some of the scan chains may be driven from random or periodic sources. With  $M > 1$ , we apply to the circuit a test set whose size is  $M$  times the size of  $T_0$ . The scan chains classified as random-testable or periodic-testable are driven from the appropriate sources. The remaining, deterministic-testable scan chains assume their original values under  $T_0$  (repeated  $M$  times). There is no increase in test data volume due to the use of  $M > 1$  since the same values are applied to the deterministic-testable scan chains repeatedly as demonstrated below.

We identify random-testable and period-testable scan chains by replacing values in  $T$  with random values or values obtained from a periodic source. One of the approaches used in [20] in order to generate hybrid tests was to replace values in deterministic tests by random values. The main difference between our approach and the one in [20] is that in [20] *different* scan chains are allowed

to assume random values under different tests. In our approach, we require that a random-testable scan chain would assume random values for the *complete* test application process, under the complete test set  $T$ . In addition, we also consider periodic-testable scan chains and we allow other types of scan chains to be considered.

To illustrate the identification of random-testable and periodic-testable scan chains, we use *s27* with  $M = 3$ . The test set  $T$  is shown in Table 2 under column  $T$ . Using  $T$ , we first identify random-testable scan chains as described next.

**Table 2: Finding random-testable scan chains for *s27***

T				ch0			
$x_0$	$x_1x_2$	$x_3x_4$	$x_5x_6$	$x_0$	$x_1x_2$	$x_3x_4$	$x_5x_6$
0	00	00	11	0	00	00	11
1	00	10	10	1	00	10	10
0	10	01	10	1	10	01	10
0	11	10	01	0	11	10	01
1	10	10	11	0	10	10	11
1	01	00	00	1	01	00	00
0	00	00	11	1	00	00	11
1	00	10	10	0	00	10	10
0	10	01	10	0	10	01	10
0	11	10	01	0	11	10	01
1	10	10	11	0	10	10	11
1	01	00	00	1	01	00	00
0	00	00	11	0	00	00	11
1	00	10	10	1	00	10	10
0	10	01	10	0	10	01	10
0	11	10	01	1	11	10	01
1	10	10	11	1	10	10	11
1	01	00	00	1	01	00	00
0	00	00	11	0	00	00	11
1	00	10	10	1	00	10	10
0	10	01	10	0	10	01	10
0	11	10	01	1	11	10	01
1	10	10	11	1	10	10	11
1	01	00	00	1	01	00	00

ch1			
$x_0$	$x_1x_2$	$x_3x_4$	$x_5x_6$
0	01	00	11
1	00	10	10
1	11	01	10
0	11	10	01
0	00	10	11
1	01	00	00
1	11	00	11
0	10	10	10
0	11	01	10
0	01	10	01
0	00	10	11
1	10	00	00
0	11	00	11
1	10	10	10
0	10	01	10
1	01	10	01
1	01	10	11
1	00	00	00

## 2.2. Random-testable scan chains

To check whether  $ch_0 = \{x_0\}$  of *s27* is random-testable, we replace the values of  $x_0$  in  $T$  by random values. The resulting test set is shown in Table 2 under column  $ch0$ . We fault simulate the faults in  $F_0$  under the new test set

$T$ . We find that all the faults are detected. Therefore, we conclude that  $ch_0$  is random-testable, and we leave the random values on  $ch_0$  in  $T$ .

Next, we consider  $ch_1 = \{x_1, x_2\}$ . Replacing the values of  $x_1$  and  $x_2$  in  $T$  by random values, we obtain the new test set  $T$  shown in Table 2 under column  $ch1$ . We fault simulate the faults in  $F_0$  under the new test set  $T$ . We find that all the faults are detected. Therefore, we conclude that  $ch_1$  is random-testable, and we leave the random values on  $ch_1$  in  $T$ . Moreover, we find at this point that  $ch_0$  and  $ch_1$  are random-testable simultaneously, since we assign random values to both  $ch_0$  and  $ch_1$  simultaneously.

After replacing  $ch_2 = \{x_3, x_4\}$  with random values we find that one fault remains undetected, and we therefore conclude that  $ch_2$  is not random-testable simultaneously with  $ch_0$  and  $ch_1$ . We restore the original values of  $x_3, x_4$ . After replacing  $ch_3 = \{x_5, x_6\}$  with random values we find that all the faults in  $F_0$  are detected. We conclude that  $ch_3$  is random-testable simultaneously with  $ch_0$  and  $ch_1$ , and we leave the random values on  $ch_3$ .

Procedure 1 below summarizes the search for random-testable scan chains. We use several techniques to speed up the fault simulation process in Procedure 1. These are described in Subsection 2.4.

**Procedure 1:** Finding random-testable scan chains

- (1) Let  $T_0$  be a compact test set for the circuit. Let  $T_0$  detect the set of faults  $F_0$ . Define a test set  $T$  where  $T_0$  is included  $M$  times.
- (2) For every scan chain  $ch_i$ :
  - (a) Replace the values of  $ch_i$  in  $T$  by pseudo-random values.
  - (b) Fault simulate  $F_0$  under  $T$ . If any fault is left undetected, restore the original values of  $ch_i$ .

Procedure 1 finds scan chains that are random-testable simultaneously since it leaves the random values in  $T$  for every scan chain identified as random-testable.

In our implementation of Procedure 1 we use software-generated random values. In hardware an *LFSR* should be used instead.

**2.3. Periodic-testable scan chains**

We use a preselected set of finite-state machines to produce periodic sequences, which may be used for driving periodic-testable scan chains. We first describe the finite-state machines, and then describe how they are used in the identification of periodic-testable scan chains.

We consider finite-state machines with two, three and four states to produce periodic sequences with periods of two, three and four. For a sequence  $\alpha$  of  $n$  bits,  $2 \leq n \leq 4$ , the finite-state machine denoted by  $PM_\alpha$  has  $n$  states, and it produces the periodic sequence  $\alpha\alpha\cdots$ . For

example, we show in Table 3 the finite-state machine  $PM_{101}$ . This machine produces the periodic sequence 101101101 $\cdots$  starting from state  $A$ . For  $n = 2$  we consider  $\alpha = 01$  and  $10$ . For  $n = 3$  we consider  $\alpha = 001, 010, \cdots, 110$ . For  $n = 4$  we consider  $\alpha = 0001, 0010, \cdots, 1110$ . In this way, we avoid the all-0 and the all-1 sequences that are not expected to allow all the faults to be detected.

**Table 3: The machine  $PM_{101}$**

PS	NS,z
A	B,1
B	C,0
C	A,1

When  $PM_\alpha$  is used for driving a scan chain, its operation is synchronized with the scan clock. Thus, every time the scan chain is shifted, the next bit of the sequence  $\alpha\alpha\cdots$  is shifted in. For example, if  $PM_{101}$  shown in Table 3 drives a scan chain of length five, the first vector scanned in (after five scan clock cycles) is 10110, the second vector scanned in (after five additional scan clock cycles) is 11011, and so on.

To check whether a scan chain  $ch_i$  of length  $L_i$  is periodic-testable, we replace the values of  $ch_i$  in  $T$  with the periodic sequence produced by  $PM_\alpha$ , for every sequence  $\alpha$  listed above. We accept the first sequence  $\alpha$  that allows  $T$  to detect all the faults in  $F_0$ . For example, for  $s27$ , after identifying random-testable scan chains, we are left with  $ch_2 = \{x_3, x_4\}$  which is not random-testable. When we consider  $\alpha = 001$ , we obtain on  $x_3x_4$  the values shown in Table 4 (the remaining inputs have been replaced by random values). This test set detects all the faults in  $F_0$ . We conclude that  $ch_2$  is periodic-testable using the sequence 001. The other scan chains are simultaneously random-testable.

**Table 4: Finding periodic-testable scan chains for  $s27$**

$x_0$	$x_1x_2$	$x_3x_4$	$x_5x_6$
0	01	00	10
1	00	10	00
1	11	01	00
0	11	00	11
0	00	10	01
1	01	01	00
1	11	00	10
0	10	10	00
0	11	01	11
0	01	00	10
0	00	10	00
1	10	01	10
0	11	00	10
1	10	10	01
0	10	01	10
1	01	00	10
1	01	10	00
1	00	01	10

Procedure 2 below summarizes the search for periodic-testable scan chains. The techniques described in Subsection 2.4 are used for speeding up the fault simulation process in Procedure 2.

**Procedure 2:** Finding periodic-testable scan chains

- (1) Let  $T$  be the test set obtained after applying Procedure 1. For every scan chain  $ch_i$  which was not found to be random-testable by Procedure 1:

For every sequence  $\alpha$ :

- (a) Replace the values of  $ch_i$  in  $T$  with values produced by  $PM_\alpha$ .
- (b) Fault simulate  $F_0$  under  $T$ . If any fault is left undetected, restore the original values of  $ch_i$ . Else, accept the current values of  $ch_i$  and do not consider additional sequences  $\alpha$  for  $ch_i$ .

**2.4. Fault simulation effort**

We use several techniques to reduce the fault simulation effort in Procedures 1 and 2.

We simulate the faults in  $F_0$  under a test set  $T$  one at a time, and stop the simulation process as soon as a fault  $f \in F_0$  is found, which is not detected by  $T$ . A single undetected fault is sufficient to reject a scan chain as a random-testable or periodic-testable scan chain. By stopping the simulation process as soon as such a fault is identified, we avoid the need to simulate the remaining faults.

We associate with every fault  $f \in F_0$  the number of times simulation of  $T$  stopped because  $f$  remained undetected. We denote this number by  $n_{fail}(f)$ . Initially,  $n_{fail}(f) = 0$  for every  $f \in F_0$ . When we simulate a new test set  $T$ , we simulate faults with high values of  $n_{fail}(f)$  first. Such faults are more likely to remain undetected by  $T$ , and the simulation process will stop as soon as an undetected fault is identified. If  $f$  is identified as undetected, we set  $n_{fail}(f) = n_{fail}(f) + 1$ .

We also associate with every fault  $f$  the index of the test in the previous test set  $T$  that detected  $f$ . We denote this index by  $t_{det}(f)$ . When we simulate  $f$  under a new test set  $T$ , we first check whether test  $t_{det}(f)$  in the new test set  $T$  detects  $f$ . In many cases, this test continues to detect the fault, and we can avoid simulating  $f$  under any additional tests.

**3. Experimental results**

We applied Procedure 1 followed by Procedure 2 to ISCAS-89 and ITC-99 benchmark circuits. As test sets  $T_0$  for ISCAS-89 benchmark circuits we used compact test sets produced by the procedure from [22]. For ITC-99 benchmark circuits we used compact test sets selected out of 100,000 random vectors. We used  $M = 1, 10$  and  $100$ . We consider numbers of internal scan chains that are

powers of two, i.e.,  $N_{CH} = 1, 2, 4, \dots$ . In addition we consider the case where the number of scan chains is equal to the number of circuit inputs. While this is not a practical scan configuration, it provides information about the numbers of inputs that are random-testable and periodic-testable, i.e., the numbers of inputs that can be assigned random or periodic values while still achieving complete fault coverage for the circuit with a limited number of tests. When we divide the circuit inputs into  $N_{CH}$  scan chains, we include approximately equal numbers of consecutive inputs in each scan chain. Thus, for a circuit with  $N_I$  inputs, we include either  $\lfloor N_I/N_{CH} \rfloor$  or  $\lceil N_I/N_{CH} \rceil$  inputs in each scan chain. We use the order by which the inputs are given in the circuit description to determine the order of the inputs in the scan chains. Thus, we do not attempt any scan chain reordering. The results are shown in Tables 5-12 in the following format.

Under column *chn* we show the number of scan chains,  $N_{CH}$ . For each value of  $M$ , we show under column *random* the number of random-testable scan chains and the fraction of random-testable scan chains. Under column *periodic* we show the number of periodic-testable scan chains and the fraction of periodic-testable scan chains. Under column *determ* we show the number of deterministic-testable scan chains and the fraction of deterministic-testable scan chains.

We do not report on very small numbers of scan chains for which no random-testable or periodic-testable scan chains are obtained. For *s35932* and *b14* with  $M = 100$ , the single scan chain obtained when  $N_{CH} = 1$  is random-testable. This implies that all the inputs can be replaced by random values. Therefore, for any number of scan chains, all the scan chains are random-testable. We do not consider larger values of  $N_{CH}$  in these cases.

The fraction of scan chains shown in Tables 5-12 is approximately equal to the fraction of inputs assigned each type of values. Therefore, the fraction of deterministic-testable scan chains is also the relative amount of test data that needs to be loaded from an external tester after eliminating the test data for the random-testable and the periodic-testable scan chains. It can be seen from Tables 5-12 that significant reductions in the amount of test data can be achieved in this way. As may be expected, larger values of  $M$  yield larger numbers of random-testable and periodic-testable scan chains, and larger reductions in test data volume.

Further reductions can be obtained by dividing deterministic-testable scan chains into subchains and identifying random-testable and periodic-testable subchains within the deterministic-testable scan chains.

To provide an indication of the run time of Procedures 1 and 2, we show in Table 13 the following information regarding the run time for *s38417* with  $M = 10$ .

We denote the run time required to check whether a single scan chain ( $N_{CH} = 1$ ) is random-testable by  $RT_0$ . We then normalize the run time of Procedures 1 and 2 to this run time. The results are shown in Table 13 for the values of  $N_{CH}$  considered in Table 10.

#### 4. Use with other compression techniques

In this section, we consider the case where the deterministic-testable scan chains of a circuit are driven from  $N_{EXT}$  external inputs through a decompressor, as shown in Figure 1.

We can determine an upper bound on  $N_{EXT}$  for the benchmark circuits considered in Section 3 based on the test sets  $T$  obtained for these circuits, as follows.

The test set  $T$  we consider is based on  $M$  repetitions of a test set  $T_0$ . Considering only the deterministic-testable scan chains, they assume  $M$  repetitions of the same scan vectors, which are derived based on  $T_0$ . We denote by  $V$  the set of scan vectors obtained for the deterministic-testable scan chains based on  $T_0$ . The number of external inputs required to encode the scan vectors in  $V$  is at most  $\lceil \log_2 |V| \rceil$  [8].

We note that the set of scan vectors  $V$  is potentially smaller than the set of scan vectors that would have been obtained if all the scan chains had been deterministic-testable. Thus, the identification of random-testable and periodic-testable scan chains contributes to a reduction in the number of external inputs. Furthermore, the identification of random-testable and periodic-testable scan chains contributes to a reduction in the number of decompressor outputs in Figure 1, which reduces the decompressor size. Further reductions in the number of scan vectors and the number of external inputs  $N_{EXT}$  can be obtained by using the procedure of [8]; however, we ignore such reductions here.

We report the upper bound on the number of external inputs  $N_{EXT}$  for benchmark circuits in Tables 14-21. We consider the case where  $M = 10$ . We consider numbers of scan chains  $N_{CH}$  that are powers of two as before. We exclude the small numbers of scan chains for which no random-testable or periodic-testable scan chains are obtained. We also exclude the two highest values of  $N_{CH}$  that result in very short scan chains (the cases excluded are the one where  $N_{CH}$  is equal to the number of circuit inputs and each scan chain is of length one, and the case where  $N_{CH}$  is the power of two closest to the number of circuit inputs where the scan chains are of lengths one and two). For every number of scan chains considered, we show in Tables 14-21 the number of deterministic-testable scan chains that remain after applying Procedure 1 followed by Procedure 2, and the fraction of deterministic-testable scan chains (these are repeated from Tables 5-12). We then show the number of scan vectors in

$V$ , and the upper bound on the number of external inputs  $N_{EXT} = \lceil \log_2 |V| \rceil$ . We also show the fraction  $N_{EXT}/N_{CH}$ .

From Tables 14-21 it can be seen that additional compression (in the form of a reduction in the number of external test inputs) can be achieved by driving the deterministic-testable scan chains through a decompressor. For example, for *s38417* with 256 scan chains, if the deterministic-testable scan chains are driven directly from external inputs, the test data volume compression obtained is  $144/256=0.56$  (which is a factor of  $256/144 = 1.78$ ). If a decompressor is used to drive the deterministic-testable scan chains, the test data is compressed by a factor of 256/10 or 25.6. Additional data compression can be achieved by using other known techniques to derive a more efficient decompressor.

#### 5. Concluding remarks

We defined random-testable and periodic-testable scan chains. A scan chain is random-testable if it is possible to achieve complete fault coverage for the circuit when the scan chain is driven from a source of pseudo-random values for a limited number of clock cycles. A scan chain is periodic-testable if it is possible to achieve complete fault coverage for the circuit when the scan chain is driven from a source that produces a periodic sequence for a limited number of clock cycles. For random-testable and periodic-testable scan chains, test data can be generated on-chip using simple sources, and there is no need to drive them from an external tester. In this way, it is possible to reduce the amount of test data that the external tester needs to store and apply to the circuit. For the remaining, deterministic-testable, scan chains that do require external test data, existing compression techniques can be used to further reduce the test data volume.

We identified random-testable and periodic-testable scan chains using a given compact deterministic test set  $T_0$ . We duplicated  $T_0$   $M$  times to obtain a test set  $T$ . To say that a scan chain is random-testable (or periodic-testable), we required that it would be possible to replace its values under  $T$  by pseudo-random values (or a periodic sequence). The results presented demonstrated that the larger the value of  $M$ , the more random-testable and periodic-testable scan chains are obtained. The amount of test data does not increase with  $M$  since the same values are applied to the deterministic-testable scan chains  $M$  times under  $T$ . Thus, there is a tradeoff between test application time and test data volume.

#### References

- [1] C.-A. Chen and S. K. Gupta, "A Methodology to Design Efficient BIST Test Pattern Generators", in Proc. 1995 Intl. Test Conf., Oct. 1995, pp. 814-823.
- [2] M. Sugihara, H. Date and H. Yasuura, "A Novel Test Methodology for Core-Based System LSIs and a Testing

- Time Minimization Problem", in Proc. Intl. Test Conf., 1998, pp. 465-472.
- [3] K. Chakrabarty and B. T. Murray, "Design of Built-In Test Generator Circuits Using Width Compression", IEEE Trans. on Computer-Aided Design, Oct. 1998, pp. 1044-1051.
- [4] K. Lee, J. Chen and C. Huang, "Using a Single Input to Support Multiple Scan Chains", in Proc. Intl. Conf. Computer-Aided Design, Nov. 1998, pp. 74-78.
- [5] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores", in Proc. Intl. Symp. on Fault-Tolerant Computing, July 1999, pp. 260-267.
- [6] I. Bayraktaroglu and A. Orailoglu, "Test Volume and Application Time Reduction Through Scan Chain Concealment", in Proc. Design Autom. Conf., June 2001, pp. 151-155.
- [7] H. Liang, S. Hellebrand and H.-J. Wunderlich, "Two-Dimensional Test Data Compression for Scan-Based Deterministic BIST", in Proc. Intl. Test Conf., Oct. 2001, pp. 894-902.
- [8] S. M. Reddy, K. Miyase, S. Kajihara and I. Pomeranz, "On Test Data Volume Reduction for Multiple Scan Chain Designs", in Proc. VLSI Test Symp., April 2002, pp. 103-108.
- [9] J. Rajski, J. Tyszer, M. Kassab, N. Kukherjee, R. Thompson, K.-H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide and J. Qian, "Embedded Deterministic Test for Low Cost Manufacturing Test", in Proc. Intl. Test Conf., 2002, pp. 301-310.
- [10] A. Jas, B. Pouya and N. A. Touba, "Virtual Scan Chains: A Means for Reducing Scan Length in Cores", in Proc. VLSI Test Symp., 2000, pp. 73-78.
- [11] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataramanan and B. Courtois, "Built-In Test for Circuits with Scan based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers", IEEE Trans. on Computers, Feb. 1995, pp. 223-233.
- [12] R. Dandapani, J. H. Patel and J. A. Abraham, "Design of Test Pattern Generation for Built-In Test", in Proc. Intl. Test Conf., 1984, pp. 315-319.
- [13] S. J. Upadhyaya and L.-C. Chen, "On-chip Test Generation for Combinational Circuits by LFSR Modification", in Proc. 1993 Intl. Conf. on Computer-Aided Design, Nov. 1993, pp. 84-87.
- [14] V. Iyengar, K. Chakrabarty, and B. T. Murray "Built-in Self Testing of Sequential Circuits Using Precomputed Test Sets," in Proc. VLSI Test Symp., April 1998, pp. 418-422.
- [15] I. Pomeranz and S. M. Reddy, "Built-In Test Sequence Generation for Synchronous Sequential Circuits Based on Loading and Expansion of Test Subsequences", in Proc. 36th Design Autom. Conf., June 1999, pp. 754-759.
- [16] L. Li and Y. Min, "An Efficient BIST Design Using LFSR-ROM Architecture", in Proc. 9th Asian Test Symp., Nov. 2000, pp. 386-390.
- [17] R. Dorsch and H.-J. Wunderlich H.-J., "Reusing Scan Chains for Test Pattern Decompression", in Proc. European Test Workshop, 2001, pp. 124-132.
- [18] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors", in Proc. Intl. Test Conf., Oct. 2001, pp. 748-757.
- [19] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth and D. Wheeler, "A SmartBIST Variant Guaranteed Encoding", in Proc. 10th Asian Test Symp., Nov. 2001, pp. 325-330.
- [20] D. Das and N. A. Touba, "Reducing Test Data Volume Using External/LBIST Hybrid Test Patterns", in Proc. Intl. Test Conf, 2000, pp. 115-122.
- [21] A. Jas, C. V. Krishna and N. A. Touba, "Hybrid BIST Based on Weighted Pseudo-Random Testing: A New Test Resource Partitioning Scheme", in Proc. VLSI Test Symp., 2001, pp. 2-8.
- [22] 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 Trans. on Computer-Aided Design, Dec. 1995, pp. 1496-1504.

**Table 5: s5378 (214 inputs, 100 tests)**

chn	M=1					
	random	periodic	determ			
16	0	0.00	1	0.06	15	0.94
32	2	0.06	1	0.03	29	0.91
64	8	0.12	4	0.06	52	0.81
128	28	0.22	19	0.15	81	0.63
214	63	0.29	54	0.25	97	0.45

chn	M=10					
	random	periodic	determ			
8	2	0.25	0	0.00	6	0.75
16	6	0.38	0	0.00	10	0.62
32	17	0.53	0	0.00	15	0.47
64	35	0.55	1	0.02	28	0.44
128	84	0.66	10	0.08	34	0.27
214	148	0.69	32	0.15	34	0.16

chn	M=100					
	random	periodic	determ			
4	2	0.50	0	0.00	2	0.50
8	6	0.75	0	0.00	2	0.25
16	13	0.81	0	0.00	3	0.19
32	26	0.81	1	0.03	5	0.16
64	57	0.89	1	0.02	6	0.09
128	116	0.91	5	0.04	7	0.05
214	191	0.89	17	0.08	6	0.03

**Table 6: s9234 (247 inputs, 111 tests)**

chn	M=1					
	random		periodic		determ	
32	1	0.03	1	0.03	30	0.94
64	4	0.06	3	0.05	57	0.89
128	21	0.16	11	0.09	96	0.75
247	62	0.25	26	0.11	159	0.64

chn	M=10					
	random		periodic		determ	
16	1	0.06	0	0.00	15	0.94
32	9	0.28	0	0.00	23	0.72
64	25	0.39	0	0.00	39	0.61
128	51	0.40	8	0.06	69	0.54
247	122	0.49	24	0.10	101	0.41

chn	M=100					
	random		periodic		determ	
16	6	0.38	0	0.00	10	0.62
32	17	0.53	0	0.00	15	0.47
64	33	0.52	0	0.00	31	0.48
128	71	0.55	3	0.02	54	0.42
247	149	0.60	25	0.10	73	0.30

**Table 7: s13207 (700 inputs, 235 tests)**

chn	M=1					
	random		periodic		determ	
32	1	0.03	1	0.03	30	0.94
64	6	0.09	7	0.11	51	0.80
128	14	0.11	14	0.11	100	0.78
256	65	0.25	49	0.19	142	0.55
512	205	0.40	114	0.22	193	0.38
700	336	0.48	179	0.26	185	0.26

chn	M=10					
	random		periodic		determ	
4	1	0.25	0	0.00	3	0.75
8	2	0.25	0	0.00	6	0.75
16	4	0.25	1	0.06	11	0.69
32	13	0.41	2	0.06	17	0.53
64	38	0.59	3	0.05	23	0.36
128	81	0.63	6	0.05	41	0.32
256	187	0.73	17	0.07	52	0.20
512	383	0.75	41	0.08	88	0.17
700	547	0.78	54	0.08	99	0.14

chn	M=100					
	random		periodic		determ	
4	1	0.25	0	0.00	3	0.75
8	3	0.38	0	0.00	5	0.62
16	8	0.50	0	0.00	8	0.50
32	22	0.69	0	0.00	10	0.31
64	50	0.78	2	0.03	12	0.19
128	107	0.84	5	0.04	16	0.12
256	218	0.85	14	0.05	24	0.09
512	445	0.87	43	0.08	24	0.05
700	623	0.89	52	0.07	25	0.04

**Table 8: s15850 (611 inputs, 97 tests)**

chn	M=1					
	random		periodic		determ	
32	1	0.03	0	0.00	31	0.97
64	4	0.06	6	0.09	54	0.84
128	20	0.16	11	0.09	97	0.76
256	60	0.23	22	0.09	174	0.68
512	167	0.33	96	0.19	249	0.49
611	202	0.33	122	0.20	287	0.47

chn	M=10					
	random		periodic		determ	
16	1	0.06	0	0.00	15	0.94
32	7	0.22	1	0.03	24	0.75
64	24	0.38	1	0.02	39	0.61
128	65	0.51	6	0.05	57	0.45
256	136	0.53	18	0.07	102	0.40
512	313	0.61	47	0.09	152	0.30
611	375	0.61	52	0.09	184	0.30

chn	M=100					
	random		periodic		determ	
16	5	0.31	0	0.00	11	0.69
32	15	0.47	0	0.00	17	0.53
64	36	0.56	0	0.00	28	0.44
128	86	0.67	2	0.02	40	0.31
256	182	0.71	10	0.04	64	0.25
512	391	0.76	29	0.06	92	0.18
611	458	0.75	40	0.07	113	0.18

**Table 9: s35932 (1763 inputs, 12 tests)**

chn	M=1					
	random		periodic		determ	
64	0	0.00	1	0.02	63	0.98
128	0	0.00	8	0.06	120	0.94
256	0	0.00	27	0.11	229	0.89
512	1	0.00	92	0.18	419	0.82
1024	96	0.09	395	0.39	533	0.52
1763	257	0.15	1107	0.63	399	0.23

chn	M=10					
	random		periodic		determ	
2	1	0.50	0	0.00	1	0.50
4	3	0.75	0	0.00	1	0.25
8	6	0.75	0	0.00	2	0.25
16	14	0.88	1	0.06	1	0.06
32	28	0.88	2	0.06	2	0.06
64	57	0.89	3	0.05	4	0.06
128	121	0.95	6	0.05	1	0.01
256	251	0.98	4	0.02	1	0.00
512	511	1.00	1	0.00	0	0.00
1024	1023	1.00	1	0.00	0	0.00
1763	1762	1.00	1	0.00	0	0.00

chn	M=100					
	random		periodic		determ	
1	1	1.00	0	0.00	0	0.00

**Table 10: s38417 (1664 inputs, 87 tests)**

chn	M=1			determ		
	random	periodic				
64	2	0.03	0	0.00	62	0.97
128	7	0.05	4	0.03	117	0.91
256	22	0.09	12	0.05	222	0.87
512	78	0.15	41	0.08	393	0.77
1024	284	0.28	123	0.12	617	0.60
1664	556	0.33	211	0.13	897	0.54

chn	M=10			determ		
	random	periodic				
64	7	0.11	1	0.02	56	0.88
128	35	0.27	8	0.06	85	0.66
256	92	0.36	20	0.08	144	0.56
512	227	0.44	42	0.08	243	0.47
1024	537	0.52	76	0.07	411	0.40
1664	933	0.56	159	0.10	572	0.34

chn	M=100			determ		
	random	periodic				
16	1	0.06	0	0.00	15	0.94
32	6	0.19	0	0.00	26	0.81
64	23	0.36	1	0.02	40	0.62
128	65	0.51	1	0.01	62	0.48
256	147	0.57	3	0.01	106	0.41
512	311	0.61	23	0.04	178	0.35
1024	673	0.66	76	0.07	275	0.27
1664	1134	0.68	191	0.11	339	0.20

**Table 12: b20 (527 inputs, 335 tests)**

chn	M=1			determ		
	random	periodic				
64	6	0.09	3	0.05	55	0.86
128	26	0.20	15	0.12	87	0.68
256	96	0.38	37	0.14	123	0.48
512	273	0.53	81	0.16	158	0.31
527	281	0.53	83	0.16	163	0.31

chn	M=10			determ		
	random	periodic				
16	1	0.06	0	0.00	15	0.94
32	4	0.12	2	0.06	26	0.81
64	22	0.34	3	0.05	39	0.61
128	61	0.48	12	0.09	55	0.43
256	150	0.59	29	0.11	77	0.30
512	367	0.72	52	0.10	93	0.18
527	378	0.72	57	0.11	92	0.17

chn	M=100			determ		
	random	periodic				
8	2	0.25	0	0.00	6	0.75
16	5	0.31	0	0.00	11	0.69
32	9	0.28	2	0.06	21	0.66
64	32	0.50	7	0.11	25	0.39
128	76	0.59	18	0.14	34	0.27
256	168	0.66	48	0.19	40	0.16
512	408	0.80	58	0.11	46	0.09
527	421	0.80	60	0.11	46	0.09

**Table 11: b14 (280 inputs, 194 tests)**

chn	M=1			determ		
	random	periodic				
16	1	0.06	2	0.12	13	0.81
32	3	0.09	5	0.16	24	0.75
64	14	0.22	13	0.20	37	0.58
128	56	0.44	26	0.20	46	0.36
256	151	0.59	52	0.20	53	0.21
280	171	0.61	56	0.20	53	0.19

chn	M=10			determ		
	random	periodic				
8	1	0.12	0	0.00	7	0.88
16	6	0.38	0	0.00	10	0.62
32	16	0.50	2	0.06	14	0.44
64	35	0.55	8	0.12	21	0.33
128	92	0.72	17	0.13	19	0.15
256	201	0.79	29	0.11	26	0.10
280	225	0.80	29	0.10	26	0.09

chn	M=100			determ		
	random	periodic				
1	1	1.00	0	0.00	0	0.00

**Table 13: Normalized run time for s38417**

chn	Proc.1	Proc.2
16	3.23	4.03
32	5.94	7.17
64	12.86	13.40
128	25.48	28.46
256	52.17	59.10
512	106.44	91.96
1024	212.88	155.81
1664	345.80	233.04

**Table 14: s5378 with compression (214 inputs, 100 tests)**

chn	determ	vectors	external		
8	6	0.75	64	6	0.75
16	10	0.62	648	10	0.62
32	15	0.47	640	10	0.31
64	28	0.44	377	9	0.14

**Table 15: s9234 with compression (247 inputs, 111 tests)**

chn	determ	vectors	external		
16	15	0.94	1396	11	0.69
32	23	0.72	884	10	0.31
64	39	0.61	444	9	0.14

**Table 16: s13207 with compression (700 inputs, 235 tests)**

chn	determ		vectors	external	
4	3	0.75	8	3	0.75
8	6	0.75	64	6	0.75
16	11	0.69	1507	11	0.69
32	17	0.53	3674	12	0.38
64	23	0.36	2296	12	0.19
128	41	0.32	1238	11	0.09
256	52	0.20	705	10	0.04

**Table 17: s15850 with compression (611 inputs, 97 tests)**

chn	determ		vectors	external	
16	15	0.94	2755	12	0.75
32	24	0.75	1829	11	0.34
64	39	0.61	960	10	0.16
128	57	0.45	485	9	0.07
256	102	0.40	291	9	0.04

**Table 18: s35932 with compression (1763 inputs, 12 tests)**

chn	determ		vectors	external	
2	1	0.50	2	1	0.50
4	1	0.25	2	1	0.25
8	2	0.25	4	2	0.25
16	1	0.06	2	1	0.06
32	2	0.06	4	2	0.06
64	4	0.06	16	4	0.06
128	1	0.01	1	1	0.01
256	1	0.00	2	1	0.00
512	0	0.00	0	0	0.00

**Table 19: s38417 with compression (1664 inputs, 87 tests)**

chn	determ		vectors	external	
64	56	0.88	2262	12	0.19
128	85	0.66	1131	11	0.09
256	144	0.56	609	10	0.04
512	243	0.47	348	9	0.02

**Table 20: b14 with compression (280 inputs, 194 tests)**

chn	determ		vectors	external	
8	7	0.88	128	7	0.88
16	10	0.62	986	10	0.62
32	14	0.44	1612	11	0.34
64	21	0.33	853	10	0.16
128	19	0.15	388	9	0.07

**Table 21: b20 with compression (527 inputs, 335 tests)**

chn	determ		vectors	external	
16	15	0.94	9132	14	0.88
32	26	0.81	5338	13	0.41
64	39	0.61	2749	12	0.19
128	55	0.43	1510	11	0.09
256	77	0.30	740	10	0.04