

# On Undetectable Faults in Partial Scan Circuits

Irith Pomeranz<sup>1</sup>  
School of Electrical & Computer Eng.  
Purdue University  
W. Lafayette, IN 47907

and Sudhakar M. Reddy<sup>2</sup>  
Electrical & Computer Eng. Dept.  
University of Iowa  
Iowa City, IA 52242

## Abstract

We provide a definition of undetectable faults in partial scan circuits under a test application scheme where a test consists of primary input vectors applied at-speed between scan operations. We also provide sufficient conditions for a fault to be undetectable under this test application scheme. We present experimental results on finite-state machine benchmarks to demonstrate the effectiveness of these conditions in identifying undetectable faults.

## 1. Introduction

Under a commonly used test application scheme for scan circuits, a test consists of a scan-in operation followed by a sequence of primary input vectors that are applied at-speed. At-speed application of primary input vectors implies that a next-state computed through the circuit logic is captured and used as a present-state for the application of the next primary input vector. A test ends with a scan-out operation. We denote a test by  $\tau = (SI, T)$ , where  $SI$  is the scan-in vector and  $T$  is the primary input sequence. We refer to this test application scheme as *scan-per-test*. In the case of a partial scan circuit, we assume that  $SI$  specifies the scanned state variables, and that the values stored in the unscanned state variables are unknown after a scan operation. This ensures that a set of tests  $\tau_1, \tau_2, \dots, \tau_m$  can be applied in any order, which facilitates test compaction. For example, one can use reverse order fault simulation to reduce the number of tests. In addition, the circuit does not need to be simulated *during* a scan operation to determine the states of the unscanned flip-flops at the end of the scan operation, nor is there a need to hold the circuit state during a scan operation. Furthermore, for efficiency reasons, when generating a test for a new fault, test generation procedures typically assume an all-unknown initial state for the unscanned flip-flops rather than a specific known state for these flip-flops. Test generation procedures that use the scan-per-test scheme were described in [1]-[4]. Commercial tools also use this test applications scheme.

A different test application scheme for partial scan circuits consists of using the scan chain to scan-in a new state before every primary input vector is applied. In this case, the option of assigning unspecified values to unscanned state variables is not viable, since these state variables will be unspecified at every time unit. Instead, the following options may be used. If the state of the unscanned flip-flops is allowed to change during scan, the fault simulation and test generation processes need to keep track of the changes in the state of these flip-flops that occur during a scan operation. For this purpose, it is necessary to use sequential circuit fault simulation over a number of time units proportionate to the number of scanned flip-flops. To avoid

this computational overhead, it is typically implied in this test application scheme that the state of the unscanned state variables is held during a scan operation. The result is that the circuit is equivalent (for the purposes of fault simulation and test generation) to a circuit where the scanned state variables are replaced by controllable inputs and observable outputs. We refer to this test application scheme as *scan-per-vector*. This test application scheme was used, for example, in the scan selection procedure of [5]. Thus, the procedure of [5] achieves the full scan fault coverage by using partial scan assuming that there is a hold mode for the unscanned state variables.

The scan-per-test scheme has several advantages over the scan-per-vector scheme, including (1) the removal of the need for holding the circuit state during scan, (2) the ability to test the circuit at-speed by applying more than one primary input vector between scan operations, and (3) a reduced test application time because of the use of fewer scan operations. However, it was shown in [4] that for partial scan circuits (where the set of scanned flip-flops is a proper subset of the circuit flip-flops), a fault which is detectable under the scan-per-vector scheme may not be detectable under the scan-per-test scheme. Because of the advantages listed above of the scan-per-test test application scheme and its use in commercial and academic tools, it is important to rigorously study the sets of undetectable faults under this scheme. Such a study was not performed in [4], and we undertake such a study in this work. Studies of undetectable faults in non-scan sequential circuits were described in [6]-[15].

We are interested only in the faults that are uniquely undetectable due to the use of partial scan. Specifically, if a fault is detectable in the non-scan sequential circuit, it is guaranteed to be detectable under partial scan. Similarly, if a fault is undetectable using full scan (the fault is combinational redundant), it is guaranteed to be undetectable under partial scan. Thus, we are interested in the undetectable faults in the combinational logic of a partial scan design that are undetectable in the non-scan sequential circuit, but are detectable using full-scan. Figure 1 demonstrates the above sets of faults. Outside the large box we have the faults that are detectable in the non-scan sequential circuit. The set of faults that are undetectable in the non-scan circuit (marked by the large box) contains the set of combinationally redundant faults (marked by the small box). Between the two boxes we have the set of faults we are interested in. These faults may be detectable or undetectable in a given partial scan circuit.

We provide for the first time a rigorous definition of undetectable faults in a partial scan circuit under the scan-per-test test application scheme. We also provide sufficient conditions for a fault in a partial scan circuit to be undetectable under this scheme. We use these conditions to identify undetectable faults in finite-state machine benchmark circuits with partial scan. The results presented provide data regarding the numbers of undetectable faults in partial scan benchmark circuits that are not due to combinational redundancy.

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.

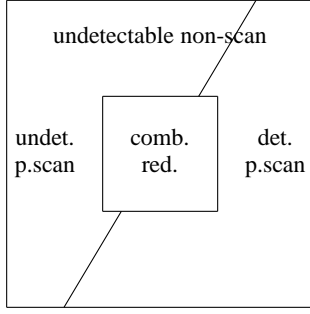


Figure 1: Target faults

## 2. Definitions

In this section, we introduce the notation used throughout this paper and provide a definition of detectable and undetectable faults in partial scan circuits under the scan-per-test scheme using tests of the form  $\tau = (SI, T)$ .

To simplify the notation, we write state vectors such that the values of the scanned state variables are given first, followed by the values of the unscanned state variables. We denote the number of scanned state variables by  $k$ , and the total number of state variables by  $K$ . For example, for a circuit with two scanned state variables and one unscanned state variable, we have  $k = 2$  and  $K = 3$ . The state 110 for this circuit implies that the scanned state variables are in state 11, and the unscanned state variable is in state 0.

We denote primary input vectors by  $t$  or  $t_i$ , primary output vectors by  $z$  or  $z_i$ , and states by  $s$  or  $s_i$ . Sets of states are denoted by  $S$  or  $S_i$ .

When we consider fault free and faulty circuits simultaneously, we use  $s_i/s_j$  to denote that the fault free circuit is in state  $s_i$  and the faulty circuit is in state  $s_j$ . Similarly, we use  $z_i/z_j$  to denote that the fault free circuit produces a primary output vector  $z_i$  and the faulty circuit produces a primary output vector  $z_j$ .

We denote by  $S_i$  the set of states  $s$  such that the values of the first  $k$  state variables (the state variables corresponding to the scanned flip-flops) are assigned the binary representation of  $i$ . For example, for a circuit with two scanned state variables and one unscanned state variable,  $S_0 = \{000, 001\}$ ,  $S_1 = \{010, 011\}$ ,  $S_2 = \{100, 101\}$  and  $S_3 = \{110, 111\}$ . We can also represent  $S_i$  as an incompletely specified state, for example,  $S_0 = 00x$ ,  $S_1 = 01x$ , and so on.

The importance of the sets of states  $S_i$  is related to the fact that if the scan-in vector is equal to the binary representation of  $i$ ,  $S_i$  is the set of possible states of the (fault free as well as faulty) circuit after scan. When we consider fault free and faulty circuits simultaneously and the scan-in vector is equal to the binary representation of  $i$ , the set of states after scan is  $S_i/S_i = \{s_p/s_q : s_p, s_q \in S_i\}$ . For example, for a circuit with two scanned state variables and one unscanned state variable, if  $i = 1$ , we have  $S_1 = \{010, 011\}$ , and  $S_1/S_1 = \{010/010, 010/011, 011/010, 011/011\}$  is the set of possible states of the fault-free/faulty circuits after scan.

We denote by  $D$  the set of states such that if the fault free and faulty circuits are brought to a state in  $D$ , the fault can be detected by a scan-out operation. A state  $s_p/s_q \in D$  satisfies the following condition. Let  $s_p \in S_i$  (i.e., the values of the first  $k$  state variables form the binary representation of  $i$  under  $s_p$ ), and let  $s_q \in S_j$  (i.e., the values of the first  $k$  state variables form the

binary representation of  $j$  under  $s_q$ ). Using this notation,  $s_p/s_q \in D$  (i.e., a fault can be detected by scanning out the state when the circuits are in state  $s_p/s_q$ ) if and only if  $i \neq j$ . For example, for a circuit with two scanned state variables and one unscanned state variable, we have the following states in  $D$ . For  $s_p \in S_0$  and  $s_q \in S_1$ , we have in  $D$  the four states covered by the partially specified state  $00x/01x$ , or  $000/010$ ,  $000/011$ ,  $001/010$  and  $001/011$ . For  $s_p \in S_0$  and  $s_q \in S_2$ , we obtain the states  $000/100$ ,  $000/101$ ,  $001/100$  and  $001/101$  in  $D$ . Similarly, we obtain in  $D$  four states for every one of  $S_0/S_1$ ,  $S_0/S_2$ ,  $S_0/S_3$ ,  $S_1/S_2$ ,  $S_1/S_0$ ,  $\dots$ ,  $S_3/S_2$ .

We denote by  $Z(T, s_p)$  the output sequence produced by the fault free circuit in response to a primary input sequence  $T$  when its initial state is  $s_p$ . We denote by  $Z_{fy}(T, s_q)$  the output sequence produced by the faulty circuit in response to a primary input sequence  $T$  when its initial state is  $s_q$ . We denote by  $s(T, s_p)$  the final state reached by the fault free circuit if the primary input sequence  $T$  is applied when its initial state is  $s_p$ . We denote by  $s_{fy}(T, s_q)$  the final state reached by the faulty circuit if the input sequence  $T$  is applied when its initial state is  $s_q$ .

We can now define detectable and undetectable faults under the scan-per-test scheme as follows. These definitions are analogous to the ones used in [8] for non-scan circuits.

**Definition 1:** A fault  $f$  is said to be detectable under the scan-per-test scheme if there exists an input sequence  $T$  and a scan-in vector  $i$  such that for every initial state  $s_p/s_q \in S_i/S_i$  either  $Z_{fy}(T, s_q) \neq Z(T, s_p)$  or  $s(T, s_p)/s_{fy}(T, s_q) \in D$ . If  $T$  and  $i$  exist,  $\tau = (i, T)$  is a test for  $f$ .

For initial states  $s_p/s_q \in S_i/S_i$  such that  $Z_{fy}(T, s_q) \neq Z(T, s_p)$ , fault effects can be observed on the primary outputs. For initial states  $s_p/s_q \in S_i/S_i$  such that  $s(T, s_p)/s_{fy}(T, s_q) \in D$ , fault effects can be observed on the scan-out vector at the end of the test.

**Definition 2:** A fault  $f$  is said to be undetectable if it is not detectable.

## 3. Sufficient conditions for identification of undetectable faults

In this section, we develop sufficient conditions for the identification of undetectable faults based on Definitions 1 and 2. In Subsection 3.1 we discuss the identification of undetectable faults using a special type of pairwise distinguishing sequences. In Subsection 3.2 we discuss the identification of undetectable faults using an iterative logic array of the circuit.

### 3.1. Using pairwise distinguishing sequences

We define a variable  $d(p, q)$  for every initial state  $s_p/s_q$  of the fault-free/faulty circuit. This variable indicates whether or not it is possible to propagate a fault effect to a primary output or to a scanned state variable starting from state  $s_p/s_q$ . We have  $d(p, q) = 1$  if there exists an input sequence  $T_{p,q}$  such that  $Z(T_{p,q}, s_p) \neq Z_{fy}(T_{p,q}, s_q)$  or  $s(T_{p,q}, s_p)/s_{fy}(T_{p,q}, s_q) \in D$ . Otherwise,  $d(p, q) = 0$ . We refer to  $T_{p,q}$  as a pairwise distinguishing sequence for states  $s_p/s_q$ . This is different from the conventional definition of pairwise distinguishing sequences since we allow the two states to be distinguished on the scanned state variables.

Using  $d(p, q)$ , the following is a necessary condition for a fault  $f$  to be detectable.

**Theorem 1:** If a fault  $f$  is detectable, there exists an  $i$  such that for every initial state  $s_p/s_q \in S_i/S_i$ ,  $d(p, q) = 1$ .

**Proof:** From the definition of a detectable fault, there exists an

input sequence  $T$  and a scan-in vector  $i$  such that for every initial state  $s_p/s_q \in S_i/S_i$ , either  $Z_{f_y}(T, s_q) \neq Z(T, s_p)$  or  $s(T, s_p)/s_{f_y}(T, s_q) \in D$ . The existence of  $T$  implies that  $d(p, q) = 1$  for every  $s_p/s_q \in S_i/S_i$ , with  $T_{p,q} = T$ .  $\square$

Note that according to the definition of  $d(p, q)$ , we may use a different sequence  $T_{p,q}$  to set  $d(p, q) = 1$  for different states  $s_p/s_q$ . Therefore,  $T_{p,q}$  may not define a test for  $f$  when used with a scan-in state  $i$ . In fact, a test may not exist even if  $d(p, q) = 1$  for every state  $s_p/s_q \in S_i/S_i$  for some  $i$ . The following example demonstrates this point.

We consider the circuit with the state table shown in Table 1. Faulty values are shown to the right of a slash in Table 1. The state names are shown on the left in Table 1. We assume that the first state variable is scanned and the second one is not scanned. Thus, we have  $S_0 = \{00, 01\} = \{s_0, s_1\}$  and  $S_1 = \{10, 11\} = \{s_2, s_3\}$ . It can be verified that  $d(2, 2) = 0$  (in addition,  $d(2, 3) = d(3, 2) = d(3, 3) = 0$ ). Thus, the fault cannot be detected after scanning in the vector  $i = 1$ . We have  $d(0, 0) = 1$  since  $s_0/s_0$  produce different output values in response to  $T_{0,0} = 0$  ( $Z(0, s_0) = 1$  and  $Z_{f_y}(0, s_0) = 0$ );  $d(0, 1) = 1$  since  $s_0/s_1$  produce different output values in response to  $T_{0,1} = 0$ ;  $d(1, 0) = 1$  since  $s_1/s_0$  produce different next state values on the scanned state variable in response to  $T_{1,0} = 1$  ( $s(1, s_1) = 01$  and  $s_{f_y}(1, s_0) = 11$ ); and  $d(1, 1) = 1$  since  $s_1/s_1$  produce different next state values on the scanned state variable in response to  $T_{1,1} = 1$ . Thus, the necessary condition of Theorem 1 is satisfied for the fault. However, the fault is not detectable since we cannot find a *single* test sequence  $T$  that will distinguish all the initial states simultaneously. Specifically, after applying the primary input vector 0, state  $s_1/s_1$  goes to state  $s_3/s_3$ , and we cannot detect the fault. If, instead, we apply the primary input vector 1, state  $s_0/s_0$  goes to state  $s_3/s_3$  and we cannot detect the fault. Thus, Theorem 1 does not provide a sufficient condition for a fault to be detectable.

**Table 1: Example fault**

	PS	NS, $z$	
		0	1
$s_0$	00	11, 1/0	11, 0
$s_1$	01	11, 0	01/11, 0
$s_2$	10	11, 0	10, 0
$s_3$	11	11, 0	11, 0

It is interesting to note that for non-scan sequential circuits, it was shown earlier [8] that a fault is detectable if and only if every pair of states  $s_p/s_q$  is distinguishable (by observing primary output sequences). In the case of partial scan circuits, the existence of pairwise distinguishing sequences (using scanned flip-flops as well as primary output sequences for observation) is a necessary condition but not a sufficient one. The example above illustrates this difference in the conditions for detectability of faults in partial scan circuits and non-scan circuits.

In a vast majority of cases where a circuit has fault-free/faulty states  $s_p/s_q$  such that  $d(p, q) = 0$ , it also has a state  $s_m$  such that  $d(m, m) = 0$ . Thus, we further simplify the necessary condition given in Theorem 1 by considering states of the form  $s_m/s_m$ .

**Corollary 1:** If a fault  $f$  is detectable, there exists an  $i$  such that for every initial state  $s_m/s_m \in S_i/S_i$ ,  $d(m, m) = 1$ .

From Theorem 1 and Corollary 1, we have the following sufficient conditions for a fault to be undetectable.

**Corollary 2:** A fault  $f$  is undetectable if for every  $i$  there exists

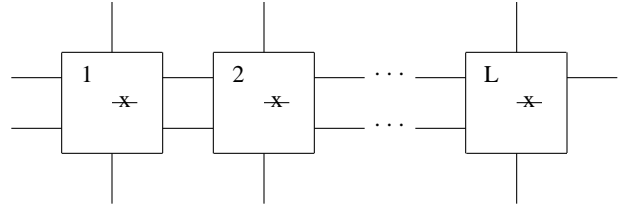
an initial state  $s_p/s_q \in S_i/S_i$  such that  $d(p, q) = 0$ .

**Corollary 3:** A fault  $f$  is undetectable if for every  $i$  there exists an initial state  $s_m/s_m \in S_i/S_i$  such that  $d(m, m) = 0$ .

### 3.2. Using an iterative logic array

One way to compute the variables  $d(p, q)$  required by Corollary 1 or the variables  $d(m, m)$  required by Corollary 3 is by using the iterative logic array of the circuit. The iterative logic array is used for test generation for scan as well as non-scan sequential circuits [16]. It is also used for identification of undetectable faults in non-scan sequential circuits [6], [7], [9]-[15].

The iterative logic array of length  $L$  of a partial scan circuit is shown in Figure 2. The top vertical lines stand for primary inputs and the bottom vertical lines stand for primary outputs. The top horizontal lines stand for scanned state variables and the bottom horizontal lines stand for unscanned state variables. In general, we inject the fault under consideration into every cell of the iterative logic array, and attempt to detect the fault under the following conditions.



**Figure 2: Iterative logic array of a partial scan circuit**

- (1) The present state variables of the first cell are controllable. In addition, the primary inputs of all the cells are controllable.
- (2) The scanned next state variables of the last cell are observable, while the unscanned next state variables of the last cell are not observable. In addition, the primary outputs of all the cells are observable.
- (3) The value of  $L$  is not restricted, and  $L$  can be increased as necessary to detect the fault or prove that it is not detectable.

If the fault is detectable in the iterative logic array for initial state  $s_m/s_m$ , then  $d(m, m) = 1$ . Thus, the iterative logic array provides a mechanism to compute  $d(m, m)$ . However, computing  $d(m, m)$  for every state  $s_m/s_m$  or even  $s_m/s_m \in S_i/S_i$  for a given  $i$  or a given  $m$  as required by Corollaries 1 or 3 may not be feasible for large circuits. To alleviate this problem, we restrict the initial states of the iterative logic array to states denoted by  $s_a/s_a$ , defined as follows.

Starting from  $s_a/s_a$ , there exists an input sequence  $T_{a,a}$  which is a pairwise distinguishing sequence for  $s_a/s_a$ , and in addition, the fault under consideration is propagated to the next state variables at every time unit until it is detected. In the iterative logic array of Figure 2, if the initial state is  $s_a/s_a$  and the input sequence is  $T_{a,a}$ , there is a 0/1 or 1/0 on an unscanned next state variable at time units 1, 2, ...,  $L-1$ , and there is a 0/1 or 1/0 on a primary output or a scanned state variable at time unit  $L$ . We refer to the sequence  $T_{a,a}$  as a *continuously propagating* pairwise distinguishing sequence.

Next, we show that if a fault  $f$  is detectable, a state  $s_a/s_a$  with a continuously propagating pairwise distinguishing sequence must exist. Thus, if it is not possible to find a state  $s_a/s_a$  with a continuously propagating pairwise distinguishing sequence,  $f$  is undetectable. The importance of this result is that the requirement for a 0/1 or 1/0 on an unscanned next state vari-

able at every time unit until  $s_a/s_a$  are distinguished facilitates the search for  $s_a$  and the input sequence  $T_{a,a}$ .

**Theorem 2:** If a fault  $f$  is detectable, there exists an initial state  $s_a/s_a$  for  $f$  with a continuously propagating pairwise distinguishing sequence, i.e., an input sequence  $T_{a,a}$  such that starting from  $s_a/s_a$ ,  $T_{a,a}$  propagates a fault effect to an unscanned state variable at every time unit except for the last one; at the last time unit,  $T_{a,a}$  propagates a fault effect to a primary output or to a scanned state variable.

**Proof:** Suppose that  $f$  is detectable. According to Corollary 1, there exists an initial state  $s_m/s_m$  such that  $d(m, m) = 1$ . From  $d(m, m) = 1$ , there exists a pairwise distinguishing sequence for  $s_m/s_m$ , i.e., a sequence  $T_{m,m}$  such that  $Z(T_{m,m}, s_m) \neq Z_{fy}(T_{m,m}, s_m)$  or  $s(T_{m,m}, s_m)/s_{fy}(T_{m,m}, s_m) \in D$ . Let the length of  $T_{m,m}$  be  $L$ , and suppose that  $T_{m,m}$  distinguishes  $s_m/s_m$  for the first time at time unit  $L$  on the primary outputs or scanned next-state variables (otherwise, it is possible to use a shorter sequence  $T_{m,m}$ ). Let the state of the circuit at time unit  $u$  under  $T_{m,m}$  be  $s_u/s_{fy,u}$ . We have  $s_u/s_{fy,u} = s_m/s_m$ . Starting from time unit  $L$ , we find the *highest* time unit  $u$  where  $s_u = s_{fy,u}$ . Such a time unit  $u$  must exist since  $s_1 = s_{fy,1} = s_m$ . From the definition of  $s_u/s_u$ , we have that  $s_{u+1} \neq s_{fy,u+1}$ ,  $\dots$ ,  $s_L \neq s_{fy,L}$  and  $s_u/s_u$  are distinguished at time unit  $L$ . Thus, the subsequence of  $T_{m,m}$  that starts at time unit  $u$  and ends at time unit  $L$  is a continuously propagating pairwise distinguishing sequence for  $s_u/s_u$ . Thus,  $s_u/s_u$  is the required state  $s_a/s_a$  and the subsequence of  $T_{m,m}$  that starts at time unit  $u$  and ends at time unit  $L$  is the required sequence  $T_{a,a}$ .  $\square$

**Corollary 4:** A fault  $f$  is undetectable if there does not exist an initial state  $s_a/s_a$  for which there is a continuously propagating pairwise distinguishing sequence  $T_{a,a}$ .

## 4. Experimental results

In this section we describe the identification of undetectable faults in partial scan benchmark circuits using Corollary 3 introduced in Section 3. As discussed earlier, we are interested in faults that are undetectable in the non-scan sequential circuit but detectable using full scan (as illustrated by Figure 1).

In all our experiments, we use gate level descriptions of finite-state machine synthesis benchmarks. We use arbitrary sets of scanned flip-flops, as follows. For a circuit with  $K$  state variables  $y_1, y_2, \dots, y_K$ , if  $k$  state variables are scanned, we assume that  $y_1, \dots, y_k$  are scanned and  $y_{k+1}, \dots, y_K$  are unscanned.

We compute pairwise distinguishing sequences  $T_{m,m}$  for states  $s_m/s_m$  by using an exhaustive search procedure. The procedure starts from  $s_m/s_m$  and explores all the state pairs reachable from  $s_m/s_m$  until one of the following conditions is satisfied. (1) The states are distinguished on a primary output. (2) The states are distinguished on a scanned next-state variable. (3) No new state pairs can be obtained. In Cases 1 and 2 we set  $d(m, m) = 1$ . In Case 3 we set  $d(m, m) = 0$ .

The results obtained are given in Table 2. After the circuit name and the number of state variables, we show in Table 2 the following numbers of faults.

- (1) The total number of faults.
- (2) The number of faults that are detectable without using scan. We compute this number based on the definition from [8].
- (3) The number of combinational redundant faults.
- (4) The number of remaining faults that are not detectable without scan and are not combinational redundant. These faults may be detectable using partial scan according to Definition 1.

**Table 2: Undetectable faults using pairwise distinguishing sequences**

circuit	sv	non				p.scan undet					
		flts	-sc det	cmb red	lft	1	2	3	4	5	6
dk16	5	532	529	2	1	1	1	1	1	0	
dvrain	6	425	424	0	1	1	1	1	1	1	0
ex4	4	176	171	0	5	5	5	5	0		
ex7	4	160	149	1	10	10	10	10	0		
keyb	5	470	468	0	2	2	2	2	2	0	
rie	5	552	545	4	3	3	3	3	3	0	
bbara	4	138	130	0	8	8	7	1	0		
bbsse	4	238	235	0	3	2	2	1	0		
bdtas	3	63	62	0	1	1	0	0			
dk512	4	124	122	0	2	2	2	1	0		
fetch	5	345	335	3	7	5	5	5	4	0	
lion9	3	62	53	3	6	6	0	0			
mark1	4	204	197	1	6	6	6	3	0		
opus	4	181	180	0	1	0	0	0	0		
train11	4	104	100	0	4	4	2	0	0		
bcount	3	112	110	2	0						
cse	4	357	355	2	0						
dk14	3	208	207	1	0						
dk15	4	151	151	0	0						
dk17	3	128	128	0	0						
dk27	3	67	67	0	0						
ex2	5	312	312	0	0						
ex3	4	153	153	0	0						
ex5	3	152	138	14	0						
ex6	3	229	229	0	0						
lion	4	40	40	0	0						
log	5	313	312	1	0						
mc	4	73	73	0	0						
shftreg	3	28	28	0	0						
tav	4	64	64	0	0						
train4	4	34	34	0	0						

(5) For  $k = 1, 2, \dots, K$ , we show the number of faults that are proved to be undetectable based on Corollary 3 if  $k$  state variables are scanned. Note that  $k = K$  corresponds to the case of full-scan. Since we are considering under column *p.scan undet* only faults that are combinational detectable, and are thus detectable under full scan, the number of undetectable faults under column  $k = K$  is always zero.

The following points can be seen from Table 2.

- (1) For the first six circuits in Table 2, full scan is necessary to detect any one of the combinational irredundant faults that are not detectable in the non-scan sequential circuit. Some of these faults may be detectable under a different partial scan selection, but they are not detectable under the partial scan selection scheme we used.
- (2) For the 16 circuits shown at the end of Table 2, all the faults are either detectable without using scan or combinational redundant. For these circuits there are no faults that can be detected by using partial scan, which are not detectable without using scan. Adding these 16 circuits to the six circuits that require full scan, we have 22 circuits for which partial scan will not allow us to detect any additional faults compared to a non-scan circuit.
- (3) The remaining nine circuits shown in the middle of Table 2 have faults that are undetectable in the non-scan circuit, but that

may be detected using partial scan without requiring full scan. Some of these circuits also have faults that require full scan to be detected.

Since the set of undetectable faults may depend on the partial scan selection, we repeated the experiment above, this time considering all possible selections of  $k$  scanned flip-flops. The most testable partial scan circuit is obtained if  $K-1$  flip-flops are scanned. Therefore, we considered  $k = K-1$  in this experiment in order to provide information about the highest testability level that can be achieved using partial scan. For each selection of  $K-1$  scanned flip-flops, we find the number of undetectable faults according to Corollary 3. The first selection will give a number of undetectable faults that matches the number under column *p.scan undet* subcolumn  $K-1$  of Table 2. Different numbers may be obtained with different selections of  $K-1$  scan flip-flops.

The results obtained for the first 15 circuits of Table 2 are given in Table 3 (the remaining circuits do not have faults that are undetectable uniquely under partial scan). After the circuit name, we repeat the number of target faults (the number of faults that are undetectable in the non-scan circuit but detectable using full scan). We then show for every selection of  $K-1$  scanned flip-flops the number of faults that are undetectable according to Corollary 3. Note that there are  $K$  such selections for a circuit with  $K$  flip-flops, and they are marked  $ps1, ps2, \dots, psK$  in Table 3.

**Table 3: Undetectable faults with  $K-1$  scan flip-flops**

circuit	left	p.scan undet					
		ps1	ps2	ps3	ps4	ps5	ps6
dk16	1	1	1	1	1		
dvrarn	1	1	1	1	1	1	1
ex4	5	5	4	5	5		
ex7	10	10	2	10	10		
rie	3	3	3	3	3	3	
bbara	8	1	2	1	2		
bbsse	3	1	3	2	2		
bbtas	1	0	0	1			
dk512	2	1	2	2	2		
fetch	7	4	6	4	4	4	
keyb	2	2	2	1	1	1	
lion9	6	0	0	6			
mark1	6	3	6	6	6		
opus	1	0	1	0	0		
train11	4	0	0	2	2		

Table 3 shows that there are variations in the numbers of undetectable faults in partial scan circuits depending on the scan selection. However, in 11 out of 15 circuits, there are faults that are combinational irredundant and that cannot be detected unless full scan is employed. This implies that, using the scan-per-test scheme, one may have to accept reduced fault coverage in partial scan designs relative to the fault coverage obtained using full scan. Thus, it is important to develop procedures to determine undetectable faults in partial scan designs when the scan-per-test scheme is employed.

## 5. Concluding remarks

We provided a definition of undetectable faults in partial scan circuits under the scan-per-test test application scheme, where a test consists of primary input vectors applied at-speed between scan operations. We also provided sufficient conditions for a fault to be undetectable under this test application scheme. The first condition was based on pairwise distinguishing sequences computed for initial state pairs. The second condition was based on a single initial state pair for which continuous propagation of the fault was required. We presented experimental results on finite-state machine benchmarks to demonstrate the effectiveness of these conditions in identifying undetectable faults.

## References

- [1] D. K. Pradhan and J. Saxena, "A Design for Testability Scheme to Reduce Test Application Time in Full Scan", in Proc. 10th VLSI Test Symp., April 1992, pp. 55-60.
- [2] S. Y. Lee and K. K. Saluja, "An Algorithm to Reduce Test Application Time in Full Scan Designs", in Proc. 1992 Intl. Conf. on Computer-Aided Design, Nov. 1992, pp. 17-20.
- [3] S. Y. Lee and K. K. Saluja, "Test Application Time Reduction for Sequential Circuits with Scan", IEEE Trans. on Computer-Aided Design, Sept. 1995, pp. 1128-1140.
- [4] I. Pomeranz and S. M. Reddy, "Simulation Based Test Generation for Scan Designs", in Proc. Intl. Conf. on Computer-Aided Design, Nov. 2000, pp. 544-549.
- [5] X. Lin, I. Pomeranz and S. M. Reddy, "Full Scan Fault Coverage With Partial Scan", in Proc. Conf. on Design Automation and Test in Europe, 1999, pp. 468-472.
- [6] V. D. Agrawal and S. T. Chakradhar, "Combinational ATPG Theorems for Identifying Untestable Faults in Sequential Circuits", in Proc. 1993 Europ. Test Conf., pp. 249-253.
- [7] K.-T. Cheng, "On Removing Redundancy in Sequential Circuits", in Proc. of 28th Design Autom. Conf., June 1991, pp. 164-169.
- [8] I. Pomeranz and S. M. Reddy, "Classification of Faults in Synchronous Sequential Circuits", IEEE Trans. on Computers, Sept., 1993, pp. 1066-1077.
- [9] M. A. Iyer and M. Abramovici, "Sequentially Untestable Faults Identified Without Search (Simple Implications Beat Exhaustive Search!)", in Proc. Intl. Test Conf., Oct. 1994, pp. 259-266.
- [10] V. D. Agrawal and S. T. Chakradhar, "Combinational ATPG Theorems for Identifying Untestable Faults in Sequential Circuits", IEEE Trans. on Computer-Aided Design, Sept. 1995., pp. 1155-1160.
- [11] M. A. Iyer, D. E. Long, and M. Abramovici, "Identifying Sequential Redundancies Without Search", in Proc. Design Autom. Conf., June 1996, pp. 259-266.
- [12] X. Lin, I. Pomeranz and S. M. Reddy, "On Finding Undetectable and Redundant Faults in Synchronous Sequential Circuits", in Proc. Intl. Conf. on Computer Design, Oct. 1998.
- [13] I. Pomeranz and S. M. Reddy, "On Identifying Undetectable and Redundant Faults In Synchronous Sequential Circuits", 12th VLSI Test Symp., April 1994, pp. 8-14.
- [14] Q. Peng, M. Abramovici and J. Savir, "MUST: Multiple-Stem Analysis for Identifying Sequentially Untestable Faults", in Proc. Intl. Test Conf., Oct. 2000, pp. 839-846.
- [15] J.-K. Zhao, J. A. Newquist and J.H. Patel, "A Graph Traversal Based Framework for Sequential Logic Implication with an Application to c-Cycle Redundancy Identification", in Proc. 14th Intl. Conf. on VLSI Design, 2001, pp. 163-169.
- [16] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.