On Application of Output Masking to Undetectable Faults in Synchronous Sequential Circuits with Design-for-Testability Logic

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

Design-for-testability (*DFT*) for synchronous sequential circuits causes redundant faults in the original circuit to be detectable in the circuit with *DFT* logic. It has been argued that such faults should not be detected in order to avoid reducing the yield unnecessarily. One way to deal with such faults is to mask (or ignore) their fault effects when they appear on the circuit outputs, without masking the detection of faults that need to be detected. To investigate the extent to which this can be accomplished, we describe a procedure for masking the effects of redundant faults of the original circuit under a given test set generated for the circuit with *DFT* logic. The procedure attempts to maximize the number of redundant faults that are masked while minimizing (or holding to zero) the number of other masked faults.

1. Introduction

The complexity of test generation for synchronous sequential circuits requires that design-for-testability (*DFT*) techniques be used to facilitate test generation [1]-[12]. Scan [1], [2] is the most popular *DFT* technique for synchronous sequential circuits. It has been noted repeatedly that redundant faults in the original circuit before *DFT* insertion become detectable after *DFT* insertion. The question that arises then is whether such faults should be detected, thus potentially reducing the yield, even though they do not affect the circuit functionality.

To facilitate the discussion of this issue, let F be the set of faults detectable in the circuit with *DFT* (in the case of full scan, these are the detectable faults in the combinational logic of the circuit). Considering the original circuit without *DFT*, the set of faults F is partitioned into two subsets: (1) The subset of detectable faults F_{OD} in the original circuit. More generally, the set F_{OD} contains the faults that can affect the functionality of the original circuit. Using the classification of [13], this set may include some undetectable faults that are not redundant. (2) The subset of redundant faults F_{OR}, which contains all the faults whose effect on the functionality of the original circuit can be neglected. The exact definition of F_{OD} and F_{OR} is not important for our discussion. What is important is that the faults in F_{OD} should be detected, while for the faults in F_{OR} one may argue that they should not be detected in order not to reduce the yield unnecessarily. In addition it is important to note that although the complexity of identifying redundant faults in synchronous sequential circuits is high, there are efficient procedures that can identify large subsets of these faults [14], [15]. Throughout the discussion we will use the fact that $F = F_{OD} \cup F_{OR}$ is the set of detectable faults in the circuit with *DFT*.

There are many reasons why faults in F_{OR} (i.e., faults that do not affect the functionality of the original circuit but are detectable in the circuit with *DFT*) should be detected. When a redundant fault is present in the circuit, it may cause a detectable fault to be missed by a test set designed under the single fault assumption [16]. Moreover, an undetectable fault may become detectable in the presence of a redundant fault and affect circuit functionality [16].

In spite of these reasons, the argument that faults in F_{OR} should not be allowed to reduce the yield is a valid one. In the industry, for high volume chips, some of the chips that fail specific structural tests are subjected to functional tests and are accepted as good if they pass the functional tests. This step is called yield recovery. Developing tests that avoid or minimize unnecessary yield reduction as proposed here may help reduce the need for functional tests, which are expensive to develop and require expensive testers for test application.

When considering the issue of detecting faults in F_{OR} due to the insertion of *DFT* logic, it is important to note that test generation is normally carried out for the circuit with *DFT* under the set of target faults F. Even if the faults in F_{OR} are excluded from F, resulting in the set of target faults $F - F_{OR} = F_{OD}$, faults out of F_{OR} may be accidentally detected in the circuit with *DFT* (we demonstrate this point later). A possible solution is to modify the test generation process so as to ensure that faults in F_{OR} are not accidentally detected by any test generated for faults in F_{OD}. However, it is possible that every test for a fault $f \in F_{OD}$ would also detect a fault $\hat{f} \in F_{OR}$, making it impossible to avoid the detection of $\hat{f} \in F_{OR}$ unless $f \in F_{OD}$ is left undetected. Moreover, a test gen-

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.

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

eration process that avoids detection of faults in F_{OR} may be too complex, defeating the purpose of inserting *DFT* logic. A solution discussed next is shown to be simple and effective. The proposed method is used for stuck-at faults. However, it is applicable to other fault models such as transition faults, path delay faults, and so on.

Suppose that a test set T is generated for the circuit with *DFT* to detect the set of faults F (or F_{OD}). We denote by R the complete response of the fault free circuit with *DFT* to the test set T. To keep the discussion independent of the *DFT* logic, we consider R as a one-dimensional array, and denote position i of R by $R[i]$. For a circuit with N_O outputs included in a single scan chain, every N_O consecutive bits in R form an output vector corresponding to a different test, and $R[i]$ is the value of output (i *mod* N_O) under test (i *div* N_O) (where *mod* is the modulo operation and *div* is integer division). Similar relationships can be written for other scan configurations or *DFT* approaches.

In the presence of a fault f, the faulty circuit with *DFT* produces a response denoted by R_f. The response R_f is given as a one-dimensional array similar to R. We denote by $I(f)$ the set of positions where R_f is different from R. We have $I(f) = \{i_1(f), i_2(f), \cdots, i_{k(f)}(f)\}$ if $R[i] \neq R_f[i]$ for $i = i_1(f), i_2(f), \cdots, i_{k(f)}(f)$.

Let $f \in F_{OR}$. It is possible to ensure that f will not be detected by the test set T if the circuit response at positions $i_1(f), i_2(f), \cdots, i_{k(f)}(f)$ is *masked*, or *ignored*. Masking or ignoring output values in positions given by $I(f)$ can affect the set of faults detectable in the original circuit, F_{OD}, in one of several ways.

(1) If every fault $\hat{f} \in F_{OD}$ can be detected in at least one position of the output response which is not in $I(f)$, i.e., $I(\hat{f}) - I(f) \neq \phi$ for every $f \in F_{OD}$, it is possible to ignore positions $I(f)$ of the output response without losing the detection of any fault in F_{OD}. In this case, it is possible to guarantee that $f \in F_{OR}$ will not be detected without any loss in fault coverage with respect to F_{OD}.

(2) If there exists a fault $\hat{f} \in F_{OD}$ that can only be detected in positions which are also included in $I(f)$, i.e., $I(\hat{f}) - I(f) = \phi$, by ignoring positions $I(f)$ of the output response we will lose the detection of $\hat{f} \in F_{OD}$. Thus, we need to decide in this case whether we prefer to detect f together with $\hat{f}$, or to lose the detection of $\hat{f}$ while masking the detection of f.

We develop in Section 2 a procedure for selecting positions that will be ignored or masked so as to minimize the number of detected faults in F_{OR}. If it is acceptable to reduce the number of detected faults in F_{OD}, then the proposed procedure can also be used for maximizing the number of detected faults in F_{OD} while simultaneously minimizing the number of detected faults in F_{OR}. Using this procedure, we study in Section 3 the range of options related to the detection of faults in F_{OR} and F_{OD}. Two of the important points we consider are the following. (1) The solution where *all* the faults in F_{OD} are detected and as many faults in F_{OR} as possible are masked. (2) The solution where all the faults in F_{OR} are masked. In Section 4 we discuss how fault dominance relations can determine whether or not the detection of a fault in F_{OR} can be masked without masking the detection of a fault in F_{OD}.

It should be mentioned that masking selected positions in test responses is already used for ignoring positions where the response of the fault free circuit is unknown. Here we use masking of selected positions of test responses in which the value of the fault free response is known in order to avoid detection of faults which should not be detected. Thus, the proposed method fits well with the existing test application practices.

The proposed procedure accepts a given test set. It is also possible to apply the proposed procedure *during* test generation to select positions that will be masked for a given test as soon as the test is generated. In this case it is possible to avoid marking as detected faults in F_{OD} that are detected on masked positions, and generate different tests for such faults.

2. Masking procedure

In this section we describe a procedure for selecting positions in the output response of a circuit that will be masked in order to avoid the detection of faults in F_{OR}. We first give an example. We then describe the general procedure.

2.1. Example

We consider a circuit with two outputs, three tests and nine faults that are detectable in the circuit with *DFT*. The faults are denoted $f_0, \cdots, f_8$. Since we have three tests and two output values for each test (corresponding to the two circuit outputs), the output response R consists of six positions numbered $0, \cdots, 5$. By performing fault simulation *with* fault dropping, we identify for every fault f the first position of the output response where it is detected. This position corresponds to the first index i where $R[i] \neq R_f[i]$. We show this information in Table 1 by placing an x in row f_j column i when f_j is detected on $R[i]$ (i.e., $R[i] \neq R_{f_j}[i]$). We assume that $F_{OD} = \{f_0, \cdots, f_6\}$ and $F_{OR} = \{f_7, f_8\}$. For the faults in F_{OR}, we need all the positions of the output response where they are detected. We obtain this information by performing fault simulation without fault dropping only for these faults. We mark the additional positions where these faults are detected by entering y's in the appropriate positions of Table 1.

Table 1: Fault detection information

		0	1	2	3	4	5
F_{OD}	f_0	x					
	f_1	x					
	f_2		x				
	f_3		x				
	f_4		x				
	f_5			x			
	f_6						x
F_{OR}	f_7	x		y			
	f_8		x	y			

To ensure that $f_7 \in F_{OR}$ is not detected, we need to mask $R[0]$ and $R[2]$. This may prevent us from detecting f_0, f_1 and f_5 if these faults cannot be detected at any other position of R. Suppose that by simulating f_0, f_1 and f_5 again, we find that f_0 can be detected on $R[3]$, f_1 can be detected on $R[4]$, and f_5 can be detected on $R[5]$. In this case, we can accept the masking of $R[0]$ and $R[2]$ without losing the detection of any fault in F_{OD}. Note that we only need to resimulate f_0, f_1 and f_5. Moreover, they can be simulated with fault dropping as long as we ignore detections on $R[0]$ and $R[2]$. The new detection information after masking $R[0]$ and $R[2]$ is shown in Table 2.

Table 2: Fault detection information after masking $R[0]$ and $R[2]$

		0	1	2	3	4	5
F_{OD}	f_0				x		
	f_1					x	
	f_2		x				
	f_3		x				
	f_4		x				
	f_5						x
	f_6						x
F_{OR}	f_7						
	f_8		x				

Next, we consider $f_8 \in F_{OR}$ assuming that $R[0]$ and $R[2]$ are already masked. To ensure that this fault is not detected, we need to mask $R[1]$. The affected faults are f_2, f_3 and f_4. We simulate these faults ignoring detections on $R[0]$, $R[1]$ and $R[2]$. Suppose that f_2 and f_3 are detected on $R[4]$, but f_4 is not detected on any other position. In this case, we can accept the masking of $R[1]$ (in addition to $R[0]$ and $R[2]$) if we accept to lose the detection of one fault in F_{OD}.

2.2. Notation

We already introduced the notation R for the one-dimensional response of the fault free circuit, R_f for the one-dimensional response of the faulty circuit in the presence of a fault f, and $I(f)$ for the positions where $R[i] \neq R_f[i]$.

We define the *masking* array M, which determines the positions of the output response that will be masked or ignored. We have $M[i] = 1$ if $R[i]$ is ignored, and $M[i] = 0$ otherwise. Ignoring or masking $R[i]$ implies that $R[i] \neq R_f[i]$ is not used for detecting f. The array M can be stored on a tester to indicate which positions of the output response should be ignored.

If f is simulated with fault dropping under a masking array M, we denote by $i_{1det}(f)$ the first position where f is detected. This is the lowest value of i where $R[i] \neq R_f[i]$ and $M[i] = 0$.

2.3. Procedure

The proposed procedure for determining the array M is described next and given formally below as Procedure 1.

The procedure uses a test set T that detects the set of faults F in the circuit with *DFT* when M is the all-0 array. In the first step of Procedure 1 we initialize M to the all-0 array. We then perform fault simulation with fault dropping to find the first position $i_{1det}(f)$ where every fault $f \in F$ is detected.

Procedure 1 performs several passes over the set of faults F_{OR}. In Pass P, it allows at most P faults in F_{OD} to be undetected due to masking of faults in F_{OR}. In Pass 0, all the faults in F_{OD} must continue to be detected, and masking is not allowed to cause any fault in F_{OD} to be undetected. The value of P is increased by one at the end of a pass. Thus, we find a solution where all the faults in F_{OD} continue to be detected, then a solution where all but one of the faults in F_{OD} continue to be detected, and so on.

To check whether the detection of $f \in F_{OR}$ can be avoided in Pass P, f is first simulated without fault dropping to find $I(f)$. Procedure 1 sets $M' = M$ and tentatively sets $M[i] = 1$ for every $i \in I(f)$. For every $\hat{f} \in F$, if $i_{1det}(\hat{f}) \in I(f)$, Procedure 1 simulates f again in an attempt to find an index i such that $R[i] \neq R_{\hat{f}}[i]$ and $M[i] = 0$. If no such index can be found for a fault $\hat{f}$, we set $i_{1det}(\hat{f}) = u$. The number of faults $\hat{f} \in F_{OD}$ for which $i_{1det}(\hat{f}) = u$ is denoted by $N_{ODM}(f)$. If $N_{ODM}(f) \leq P$, the number of undetected faults in F_{OD} is acceptable for Pass P. In this case, the detection of f is avoided and the new array M is accepted. Otherwise, M' is restored into M and $f \in F_{OR}$ remains detected. If $i_{1det}(\hat{f}) = u$ for a fault $\hat{f} \in F_{OR}$ other than f, then detection of $\hat{f}$ is avoided without having to consider $\hat{f}$ explicitly.

The following shortcuts are implemented in Procedure 1. Suppose that in Pass P no fault $f \in F_{OR}$ can be masked. This implies that $N_{ODM}(f) > P$ for every $f \in F_{OR}$. Let $N_{ODM}^{\min}$ be the smallest value of $N_{ODM}(f)$ obtained for any $f \in F_{OR}$. Then no fault $f \in F_{OR}$ can be masked until $P = N_{ODM}^{\min}$. Instead of incrementing P by one, we increment P to $N_{ODM}^{\min}$ at the end of such a pass. This shortcut is not applied if a fault $f \in F_{OR}$ is masked in Pass P. The reason is that the value of $N_{ODM}(\hat{f})$ for a fault $\hat{f}$ considered before f may not be correct after f is masked. Therefore, if at least one fault is masked in Pass P we increment P by one at the end of the pass.

For every $f \in F_{OR}$, we initialize $N_{ODM}(f)$ to zero at the beginning of Procedure 1, and we update it every time f is considered and a new value is available. We note that $N_{ODM}(f)$ can never go down as more and more output values are masked during Procedure 1. Therefore, if the current value of $N_{ODM}(f)$ available for a fault $f \in F_{OR}$ is such that $N_{ODM}(f) > P$, there is no need to consider fault f in Pass P. This allows us to avoid consideration of faults in certain passes of Procedure 1 if these faults are guaranteed to remain detected at the end of the pass.

Procedure 1: Masking output values

(1) Let T be the given test set that detects the set of faults F in the circuit with *DFT*. Set $M[i] = 0$ for every position i of the output response. Perform fault simulation with fault dropping of the faults in F under T in the circuit with *DFT* and find for every fault $f \in F$ the first posi-

tion $i_{1det}(f)$ where it is detected by T. Set $N_{ODM}(f) = 0$ for every $f \in F_{OR}$.

(2) Set $P = 0$.

(3) For every $f \in F_{OR}$ if $i_{1det}(f) \geq 0$ and $N_{ODM}(f) \leq P$:

(a) Set $M' = M$. For every $\hat{f} \in F$, set $i'_{1det}(\hat{f}) = i_{1det}(\hat{f})$.

(b) Perform fault simulation without fault dropping of f to find $I(f)$.

(c) Set $M[i] = 1$ for every $i \in I(f)$.

(d) For every fault $\hat{f} \in F$ such that $M[i_{1det}(\hat{f})] = 1$, perform fault simulation with fault dropping of $\hat{f}$, avoiding detection on positions i where $M[i] = 1$, and update $i_{1det}(\hat{f})$ ($i_{1det}(\hat{f}) = u$ if $\hat{f}$ is not detected, otherwise $i_{1det}(\hat{f}) \geq 0$ is an unmasked position where $\hat{f}$ is detected).

(e) Update the number $N_{ODM}(f)$ of faults $\hat{f} \in F_{OD}$ such that $i_{1det}(\hat{f}) = u$.

(f) If $N_{ODM}(f) > P$:
Set $M = M'$. For every $\hat{f} \in F$, set $i_{1det}(\hat{f}) = i'_{1det}(\hat{f})$.

(4) If any fault $f \in F_{OR}$ was masked is pass P, set $P = P+1$.
Else set $P = N_{ODM}^{\min} = \min\{N_{ODM}(f): f \in F_{OR}, i_{1det}(f) \geq 0\}$.

(5) If there is at least one fault $f \in F_{OR}$ such that $i_{1det}(f) \geq 0$ go to Step 3.

Procedure 1 requires the following fault simulation effort. (1) Fault simulation with fault dropping of F in Step 1. (2) Fault simulation without fault dropping for some of the faults in F_{OR} to determine $I(f)$ in Step 3(b) of every pass. In the worst case, all the faults in F_{OR} will be simulated without fault dropping. In practice, if masking of a fault $f \in F_{OR}$ results in the masking of another fault $\hat{f} \in F_{OR}$, fault simulation without fault dropping is avoided for $\hat{f}$. In addition, if $N_{ODM}(f) > P$, f is not simulated in Pass P. It is also possible to compute $I(f)$ once for every $f \in F_{OR}$ and store it to avoid having to recompute it. (3) Fault simulation with fault dropping of faults for which the positions where they are first detected are masked. In the worst case, all the faults will be simulated with fault dropping in every pass. In practice, only a small subset of the faults is simulated in every pass.

3. Experimental results

We applied Procedure 1 to ISCAS-89 and ITC-99 benchmark circuits. For the set of faults F we use the set of faults detectable assuming full scan. For F_{OD} we use the faults detectable assuming that the synchronous sequential circuit can be reset to the all-0 state. The remaining faults in F are included in the set F_{OR}. It should be noted that the faults, which are undetectable assuming the all-0 initial state, are also undetectable assuming an unknown initial state. For $s35932$ we obtain $F_{OR} = \phi$ and we do not consider this circuit.

For the test set T we initially use a conventional, compacted test set that detects all the faults in F (i.e., all the detectable faults of the full scan circuit). Later we will consider other types of test sets in order to demonstrate the effects of the test set on the results.

In Table 3 we show the following parameters for every circuit using a conventional, compacted test set.

The number of tests in T, denoted by N_T.

The number of output values obtained under T, denoted by N_{OV}. For a circuit with N_O outputs, $N_{OV} = N_T N_O$.

The numbers of faults in F, F_{OD} and F_{OR}, denoted by N_F, N_{OD} and N_{OR}, respectively.

For several passes P of Procedure 1 (shown in separate columns under $\%M, NORD, NODM$) we show the following parameters.

The percentage of masked output values, denoted by $\%M$.

The number of faults in F_{OR} that are still detected (i.e., faults $f \in F_{OR}$ with $i_{1det}(f) \geq 0$), denoted by N_{ORD}.

The number of faults in F_{OD} that are masked (i.e., faults $f \in F_{OD}$ with $i_{1det}(f) = u$), denoted by N_{ODM}. This number represents the loss in detection of faults in F_{OD}, which should be detected.

The first column under $\%M, NORD, NODM$ always corresponds to 0% masking, resulting in $N_{ORD} = N_{OR}$ and $N_{ODM} = 0$. In this case, all the faults in F_{OD} as well as all the faults in F_{OR} are detected. The second column under $\%M, NORD, NODM$ always corresponds to Pass $P = 0$ where $N_{ODM} = 0$, but as many of the faults as possible out of F_{OR} are masked. This column corresponds to the case where Procedure 1 masks the detection of a maximal number of faults in F_{OR} without losing the detection of any fault in F_{OD}. The third column corresponds to the first Pass $P \geq 1$ that allows us to mask additional faults in F_{OR} at the cost of masking faults in F_{OD}. The last column corresponds to $N_{ORD} = 0$. In this case, all the faults in F_{OR} are masked. There may be other results for values of $P \geq 1$, which are omitted.

For example, for $s298$, F_{OR} contains 35 faults. In Pass $P = 0$, we mask the detection of 32 of these faults without losing the detection of any fault in F_{OD} by masking 8.75% of the output values in R. The result is $N_{ORD} = 3$ faults in F_{OR} that continue to be detected. No originally detected faults in F_{OD} are masked, i.e., $N_{ODM} = 0$. In Pass $P = 1$, at the cost of masking one fault out of F_{OD}, the number of detected faults in F_{OR} is reduced to one. The percentage of masked output values is 9.38%. In Pass $P = 2$, at the cost of masking two faults out of F_{OD}, all the faults in F_{OR} are masked. The percentage of masked output values is 9.58%.

The second column under $\%M, NORD, NODM$ in Table 3 demonstrates that large numbers of faults in F_{OR} can be masked while still detecting all the faults in F_{OD}. The tradeoff between the number of masked faults in F_{OR} and the number of masked faults in F_{OD} is also demonstrated by Table 3. For several circuits (s953 and s1196) it is possible to mask all the faults in F_{OR} without masking any fault in F_{OD}.

The run time of Procedure 1 in seconds on a Sun U80 Workstation is shown in Table 4. Our implementation of Procedure 1 does not use some of the speedup

techniques included in the description of the procedure. The most notable technique missing from our implementation is the observation that only a fault f such that $i_{1det}(f)$ is masked needs to be simulated. Nevertheless, the run time is reasonable for all the circuits considered.

If Procedure 1 is applied using a non-compacted test set, we may be able to mask more faults in F_{OR} without masking any faults in F_{OD}. This is due to the fact that a non-compacted test set tends to detect fewer faults with each test, and as a result, it is easier to mask a fault without masking any other fault. Similarly, an n-detection test set for $n > 1$ will provide more tests for every fault and it is likely to allow us to mask faults while leaving unmasked output values to detect other faults, which should not be masked. We show the results of Procedure 1 using compacted 10-detection test sets in Table 5. We only consider several circuits where the conventional compacted test set required large numbers of faults in F_{OD} to be masked in order to mask all the faults in F_{OR}.

From Tables 5 it can be seen that 10-detection test sets are effective in allowing faults in F_{OR} to be masked without masking faults in F_{OD}. For example, in the case of s1423, it is possible to mask all the faults in F_{OR} without masking any fault in F_{OD}. Other circuits for which 10-detection test sets allowed all the faults in F_{OR} to be masked without masking any fault in F_{OD} are s344, s382, s820 and s1488.

All the test sets considered so far are generated for all the faults in F, including the faults in F_{OD} and the faults in F_{OR}. Next, we consider test sets generated by targeting only the faults in F_{OD}. Such test sets detect faults in F_{OR} only due to accidental detection by some tests generated for faults in F_{OD}. The results for such test sets are shown in Tables 6 and demonstrate the following points. (1) Accidental detections cause faults in F_{OR} to be detected even though only faults in F_{OD} are targeted. However, fewer faults out of F_{OR} are detected than when all the faults in F are targeted. (2) Since fewer faults are detected out of F_{OR}, fewer values need to be masked in order to prevent their detection. As a result, it is possible to mask all the faults in F_{OR} while masking fewer faults in F_{OD}.

4. Fault dominance relations

A fault f_2 is said to *dominate* a fault f_1 if every test that detects f_1 also detects f_2 on the same primary outputs. Note that this definition is different from the conventional definition [16] as it requires that the faults be detected on the same outputs. If f_2 dominates f_1, then masking of f_2 will imply masking of f_1.

We are interested in fault dominance relations where $f_2 \in F_{OR}$ and $f_1 \in F_{OD}$. If f_2 dominates f_1, $f_2 \in F_{OR}$ and $f_1 \in F_{OD}$, then when we mask f_2 we will also mask f_1. Thus, dominance relations can provide the following lower bounds. (1) A lower bound on the number of faults in F_{OD} that will be masked when masking a given fault $f_2 \in F_{OR}$. This is the number of faults in F_{OD} that are dominated by f_2. (2) A lower bound on the number of faults in F_{OD} that will be masked when masking all the faults in F_{OR}. This is the number of faults in F_{OD} that are dominated by at least one fault in F_{OR}.

Dominance relations as defined here are available for some of the circuits considered here. Based on these dominance relations we found that faults in F_{OD} are never dominated by faults in F_{OR} for these circuits. Thus, theoretically, it is possible to derive a test set where it will be possible to mask all the faults in F_{OR} without masking any faults in F_{OD}. Our experiments with 10-detection test sets confirm this point.

5. Concluding remarks

It has been argued that a redundant fault that becomes detectable due to the insertion of *DFT* logic should not be detected in order to prevent unnecessary yield loss. One way to deal with redundant faults that become detectable due to *DFT* logic is to mask their fault effects that reach an output. This should be done as much as possible without masking the fault effects of faults that need to be detected. To investigate the extent to which this can be accomplished, we described a procedure for masking the effects of redundant faults under a given test set. The procedure attempts to maximize the number of redundant faults that are masked while minimizing the number of other masked faults. Several points were of particular interest in this study. (1) The point where all the originally detectable faults are detected and as many originally redundant faults as possible are masked. (2) The point where all the originally redundant faults are masked. The proposed procedure allowed us to explore the range of options between these two points.

References

[1] M. J. Y. Williams and J. B. Angell, "Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic", IEEE Trans. on Computers, 1973, pp. 46-60.

[2] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability", in Proc. 14th Design Autom. Conf., June 1977, pp. 462-468.

[3] S. M. Reddy and R. Dandapani, "Scan design using standard flip-flops", IEEE Design & Test, Feb. 1987, pp. 52-54.

[4] M. Abramovici, P. S. Parikh, B. Mathew and D. G. Saab, "On Selecting Flip-Flops for Partial Reset", in Proc. Intl. Test Conf., 1993, pp. 1008-1012.

[5] V. Chickermane, E. M. Rudnick, P. Banerjee and J. H. Patel, "Non-Scan Design-for-Testability Techniques for Sequential Circuits", in Proc. 30th Design Autom. Conf., June 1993, pp. 236-241.

[6] I. Pomeranz and S. M. Reddy, "On the Role of Hardware Reset in Synchronous Sequential Circuit Test Generation", IEEE Trans. on Computers, Sept. 1994, pp. 1100-1105.

[7] P. Parikh and M. Abramovici, "On Combining Design for Testability Techniques", in Proc. 1995 Intl. Test Conf., Oct. 1995, pp. 423-429.

[8] D. K. Das and B. B. Bhattacharya, "Testable Design of Non-Scan Sequential Circuits using Extra Logic", in Proc. Asian Test Symp., Nov. 1995, pp. 176-182.

[9] I. Pomeranz and S. M. Reddy, "On the Use of Fully Specified Initial States for Testing of Synchronous Sequential Circuits", IEEE Trans. on Computers, Feb. 2000, pp. 175-182.

[10] X. Dong, X. Yi and H. Fujiwara, "Non-Scan Design for Testability for Synchronous Sequential Circuits Based on Conflict Analysis", in Proc. Intl. Test Conf., 2000, pp. 520-529.

[11] H. Fujiwara, "A New Class of Sequential Circuits with Combinational Test Generation Complexity", IEEE Trans. on Computers, Sept. 2000, pp. 895-905.

[12] M. Abramovici, X. Yu and E. M. Rudnick, "Low-Cost Sequential ATPG with Clock-Control DFT", in Proc. 39th Design Autom. Conf., June 2002, pp. 243-248.

[13] I. Pomeranz and S. M. Reddy, "Classification of Faults in Synchronous Sequential Circuits", IEEE Trans. on Computers, Sept. 1993, pp. 1066-1077.

[14] M. Iyer and M. Abramovici, "FIRE: A Fault-Independent Combinational Redundancy Identification Algorithm", IEEE Trans. on VLSI Systems, June 1996, pp. 295 -301.

[15] 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, pp. 498-503.

[16] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1995.

Table 4: Run time (sec, Sun U80)

circuit	time
s298	0.17
s344	0.06
s382	0.22
s400	0.21
s526	2.71
s641	1.87
s820	2.44
s953	0.58
s1196	0.25
s1423	3.91
s1488	5.13
s5378	1487.03
b03	3.67
b04	14.46
b09	2.43
b10	1.58
b11	14.06
b14	992.51
b20	8111.46

Table 3: Results using conventional compact test sets

circuit	NT	NOV	NF	NOD	NOR	%M,NORD,NODM			
s298	24	480	308	273	35	0.00,35,0	8.75,3,0	9.38,1,1	9.58,0,2
s344	15	390	342	335	7	0.00,7,0	2.31,1,0	5.38,0,4	
s382	25	675	399	379	20	0.00,20,0	8.44,1,0	9.48,0,4	
s400	24	648	415	395	20	0.00,20,0	7.56,1,0	8.18,0,2	
s526	50	1350	554	466	88	0.00,88,0	10.81,11,0	11.11,9,1	12.15,0,13
s641	22	946	467	408	59	0.00,59,0	10.68,12,0	10.78,10,1	13.42,0,12
s820	94	2256	850	815	35	0.00,35,0	1.64,2,0	1.68,0,1	
s953	76	3952	1079	1069	10	0.00,10,0	0.68,0,0		
s1196	136	4352	1242	1239	3	0.00,3,0	0.16,0,0		
s1423	26	2054	1501	1468	33	0.00,33,0	2.73,8,0	3.21,6,1	4.48,0,12
s1488	101	2525	1486	1446	40	0.00,40,0	2.02,2,0	2.14,1,1	2.18,0,2
s5378	100	22800	4563	3697	866	0.00,866,0	21.57,69,0	21.70,66,1	24.43,0,122
b03	43	1462	452	336	116	0.00,116,0	30.44,11,0	30.71,9,1	32.56,0,10
b04	97	7178	1344	1171	173	0.00,173,0	17.68,1,0	17.72,0,1	
b09	44	1276	420	343	77	0.00,77,0	19.12,10,0	19.67,7,1	20.61,0,11
b10	82	1886	512	471	41	0.00,41,0	8.48,4,0	8.75,3,1	9.65,0,5
b11	107	3852	1078	999	79	0.00,79,0	14.54,7,0	14.64,6,1	15.89,0,12
b14	275	82775	8936	8797	139	0.00,139,0	9.37,2,0	9.38,0,3	
b20	510	263160	20797	20501	296	0.00,296,0	10.96,1,0	10.96,0,1	

Table 5: Results using 10-detection test sets

circuit	NT	NOV	NF	NOD	NOR	%M,NORD,NODM			
s526	493	13311	554	466	88	0.00,88,0	13.13,1,0	13.24,0,1	
s641	212	9116	467	408	59	0.00,59,0	13.57,4,0	13.76,2,1	13.87,0,2
s1423	233	18407	1501	1468	33	0.00,33,0	6.54,0,0		
s5378	989	225492	4563	3697	866	0.00,866,0	23.09,18,0	23.18,17,1	23.53,0,20

Table 6: Results targeting only F_{OD}

circuit	NT	NOV	NF	NOD	NOR	%M,NORD,NODM			
s298	32	640	290	273	17	0.00,17,0	3.12,0,0		
s344	35	910	342	335	7	0.00,7,0	2.97,0,0		
s382	42	1134	393	379	14	0.00,14,0	4.41,1,0	5.56,0,2	
s400	48	1296	412	395	17	0.00,17,0	4.01,1,0	4.71,0,4	
s526	67	1809	527	466	61	0.00,61,0	6.25,2,0	6.30,1,1	6.47,0,2
s641	69	2967	453	408	45	0.00,45,0	8.09,3,0	8.83,1,1	8.86,0,3
s820	139	3336	826	815	11	0.00,11,0	0.36,0,0		
s953	117	6084	1073	1069	4	0.00,4,0	0.15,1,0	0.30,0,1	
s1196	197	6304	1242	1239	3	0.00,3,0	0.16,0,0		