N-Detection Under Transparent-Scan

Irith Pomeranz *
School of ECE
Purdue University
W. Lafayette, IN 47907

Abstract

We study the quality of test sequences under a test application scheme called transparent-scan as n-detection test sequences. We obtain transparent-scan sequences from combinational test sets. We show that for the same number of clock cycles required to apply a compact single-detection combinational test set, a transparent-scan sequence detects faults more times than the combinational test set. We note that a transparent-scan sequence based on a combinational test set contains unspecified values. We consider several procedures for specifying the unspecified values of the transparent-scan sequence, and study their effects. We also study the extension of a transparent-scan test sequence into an n-detection test sequence that detects every target fault at least n times.

Categories & Subject Descriptors: B.8.1 Reliability, Testing, and Fault-Tolerance

General Terms: Reliability

Keywords: n-detection test sets, scan design, test generation.

1. Introduction

Test sets that contain several different tests to detect each target fault, also referred to as n-detection test sets, were shown to be effective in detecting unmodeled faults and defects [1]-[9]. In this work we study the quality of test sequences under a test application scheme called transparent-scan as n-detection test sequences.

The transparent-scan approach was proposed in [10], and was shown to achieve higher levels of compaction than other test compaction approaches for scan circuits. It was also found to be necessary in [11] for testing critical paths in a microprocessor that uses partial scan. Under the transparent-scan approach, the distinction that typically exists between scan operations and functional clock cycles is eliminated. This is achieved by using the scan select and scan chain inputs as regular primary inputs of the circuit, and using the scan chain outputs as regular primary outputs of the circuit during test generation and test compaction. The result is a synchronous sequential circuit with the extra controllability and observability provided by scan.

A test under the transparent-scan approach is a sequence T of primary input vectors, which assign values to the original primary inputs of the circuit, to the scan select input, and to the scan chain inputs. An output vector includes values corresponding to the original primary outputs of the circuit, and to the scan chain outputs. This unified view of the original primary inputs (outputs) and the scan inputs (outputs) provides complete flexibility in interleaving scan operations and functional clock cycles, and results in high levels of compaction.

The *test translation* method was proposed in [10] for obtaining an initial test sequence under the transparent-scan approach. Under this method, a combinational test set C (a test set C for the combinational logic of the circuit) is translated into a test sequence T under the transparent-scan approach by translating a scan operation of C into a sequence of primary input vectors where the scan select is high, and translating a functional clock cycle of C into a primary input vector where the scan select is low. In [10], the translated test sequence T is compacted by a static test compaction procedure for synchronous sequential circuits. In this work, we will focus on the quality of T (before compaction) as an n-detection test sequence. Several properties of a transparent-scan test sequence T, which is translated from a combinational test set C, are important.

The number of clock cycles required for applying T (before compaction) is equal to the number of clock cycles required for applying C. Nevertheless, T provides more options for detecting faults, as we demonstrate later.

The test sequence T obtained after translation of a combinational test set C is incompletely specified. Even as an incompletely specified test sequence, T may detect faults more times than C. By specifying the unspecified values of T, the qualify of T as an n-detection test sequence can be improved even further.

The paper is organized as follows. In Section 2 we review the test translation method from [10] for obtaining a transparent-scan test sequence. In Section 3 we provide definitions related to the quality of a test set as an n-detection test set. We then consider the quality of transparent-scan test sequences as n-detection test sequences. We describe three procedures for specifying the unspecified values of a transparent-scan sequence so as to improve its quality. Experimental results are presented in Section 4. In Section 5 we study the possibil-

* Research supported in part by SRC Grant No. 2004-TJ-1244.

DAC 2005, June 13-17, 2005, Anaheim, California, USA.

ity of extending a transparent-scan sequence into an n-detection test sequence that detects every target fault at least n times.

2. Transparent-scan

We demonstrate the test translation method under the transparent-scan approach by considering ISCAS-89 benchmark circuit $s27$. The original circuit (without scan) has four primary inputs a_0, a_1, a_2, a_3, and three state variables. A combinational test set C for the circuit is shown in Table 1. Each combinational test vector $c_i \in C$ is divided into a subvector c_{iI} that defines the values of the original primary inputs of the circuit, and a subvector c_{iS} that defines the values of the state variables. We denote the scan select input by s_{sel}, and the scan chain input by s_{inp}. We assume that the circuit has a single scan chain, and that the scan chain is shifted from left to right. An input vector of a transparent-scan sequence consists of the values of the original primary inputs of the circuit, followed by the values of s_{sel} and s_{inp} (in this order).

Table 1: Combinational test set C for $s27$

i	c_{iI}	c_{iS}
0	0000	011
1	1001	010
2	0100	110
3	0111	001
4	1101	011
5	1010	000

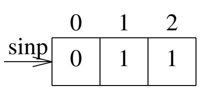

Figure 1: Single scan chain

To apply the combinational test vector c_0 of $s27$, we need the following primary input vectors. We first need to hold $s_{sel} = 1$ for three clock cycles while scanning in the state subvector $c_{0S} = 011$. Due to the assumption that scan chains are shifted to the right, this requires us to set $s_{inp} = 1$ in the first clock cycle, $s_{inp} = 1$ in the second clock cycle, and $s_{inp} = 0$ in the third clock cycle, as demonstrated by Figure 1. We obtain the input vectors $a_0 a_1 a_2 a_3 s_{sel} s_{inp}$ = xxxx11, xxxx11 and xxxx10, where an x stands for an unspecified value. Note that we list the values of the four original primary inputs first, followed by s_{sel} and s_{inp}. We then need to hold $s_{sel} = 0$ while applying $c_{0I} = 0000$ to the original primary inputs of the circuit. This is done by the input vector $a_0 a_1 a_2 a_3 s_{sel} s_{inp}$ = 00000x. Finally, we need to hold $s_{sel} = 1$ for three clock cycles in order to scan out the final state. The resulting transparent-scan sequence is T = xxxx11 xxxx11 xxxx10 00000x xxxx1x xxxx1x xxxx1x.

While scanning out the final state of c_0, we scan-in the state $c_{1S} = 010$ of c_1. We then hold $s_{sel} = 0$ while applying $c_{1I} = 1001$ to the original primary inputs of the circuit. Continuing in the same way to apply the remaining test vectors in Table 1, we obtain the transparent-scan sequence shown in Table 2 under column T_x.

It is important to note that the test sequence T_x is incompletely specified. Unspecified values are obtained since the combinational test set C does not specify primary input vectors during scan operations, and it does not specify the value of s_{inp} during a functional clock cycle. Even with the incompletely specified values, T detects all the faults detected by the combinational test set C.

Table 2: Transparent-scan test sequence for $s27$

u	T_x	T_{rnd}	CI_ind	T_{cpi}
0	xxxx11	001111	2	010011
1	xxxx11	110111	4	110111
2	xxxx10	001110	3	011110
3	00000x	000000	-	000000
4	xxxx10	011110	1	100110
5	xxxx11	110011	1	100111
6	xxxx10	011110	5	101010
7	10010x	100100	-	100100
8	xxxx10	110010	0	000010
9	xxxx11	010011	3	011111
10	xxxx11	110111	0	000011
11	01000x	010001	-	010001
12	xxxx11	000111	0	000011
13	xxxx10	110110	1	100110
14	xxxx10	100110	5	101010
15	01110x	011100	-	011101
16	xxxx11	010011	0	000011
17	xxxx11	011111	0	000011
18	xxxx10	111010	5	101010
19	11010x	110100	-	110101
20	xxxx10	101110	5	101010
21	xxxx10	000010	3	011110
22	xxxx10	000010	3	011110
23	10100x	101001	-	101001
24	xxxx1x	101011	3	011110
25	xxxx1x	001111	0	000011
26	xxxx1x	010111	2	010010

It is also important to note that the length of the test sequence T is equal to the number of clock cycles required for applying the combinational test set C from which T is obtained.

3. N-detections

In this section we discuss the quality of a transparent-scan sequence as an n-detection test set. We first provide a definition of this quality for a combinational test set and for a transparent-scan sequence. We then discuss the use of the unspecified values in T_x to improve the quality of the test sequence.

3.1. Quality definitions

The quality of an arbitrary combinational test set C as an n-detection test set can be defined as follows.

We perform n-detection fault simulation of C. Under this process, a fault is dropped only after it is detected n times, by n test vectors in C. We denote by $n_{det}(f)$ the number of times a fault f is detected at the end of this fault simulation process. We denote by F the set of faults that are detected at least once by C. For a test set C_i we use the notation $n^i_{det}(f)$ for the number of times f is detected by C_i, and F^i for the set of faults detected at least once by C_i.

A simple measure of the quality of C as an n-detection test set is obtained by using the total (or the average) number of detections under C. The total number of detections is equal to $\sum\{n_{det}(f) : f \in F\}$. The average number of detections is equal to $\sum\{n_{det}(f) : f \in F\} / |F|$.

However, the total (or average) number of detections does not take into account the following situation. Consider a circuit with three faults f_1, f_2 and f_3. Let $n = 5$. Suppose that a test set C_1 results in $n^1_{det}(f_1) = 1$, $n^1_{det}(f_2) = 2$, and $n^1_{det}(f_3) = 5$. Suppose that a test set C_2 results in $n^2_{det}(f_1) = 1$, $n^2_{det}(f_2) = 3$, and $n^2_{det}(f_3) = 4$. The total number of detections in both cases is eight. However, defects associated with f_2, which is detected twice by C_1, are more likely to be detected by C_2, where f_2 is detected three times. The reduction in the number of detections for f_3 from five to four is less likely to affect the overall quality of C_2. The resulting metric considers the numbers of detections of the faults as follows.

For $m = 1,2, \cdots ,n$, we denote by F_m the subset of faults in F that are detected m times by C, i.e., $F_m = \{f \in F : n_{det}(f)=m\}$. We denote the size of F_m by $|F_m|$. For a test set C_i we use the notation $F^i_m = \{f \in F^i : n^i_{det}(f)=m\}$. In comparing two test sets C_1 and C_2 with subsets of faults $F^1_1, F^1_2, \cdots, F^1_n$ and $F^2_1, F^2_2, \cdots, F^2_n$, respectively, such that $F^1 = F^2$, we consider the first value of m for which $|F^1_m| \neq |F^2_m|$. We say that C_1 is of higher quality if $|F^1_m| < |F^2_m|$. We say that C_2 is of higher quality if $|F^2_m| < |F^1_m|$.

For example, for C_1 and C_2 above we have $F^1_1 = \{f_1\}$ and $F^2_1 = \{f_1\}$; $F^1_2 = \{f_2\}$ and $F^2_2 = \phi$. The first value of m for which $|F^1_m| \neq |F^2_m|$ is $m = 2$, and we have $|F^2_m| < |F^1_m|$. We conclude that C_2 is of higher quality.

To define the quality of a test sequence T under the transparent-scan approach, we perform n-detection fault simulation of T. During this process, $n_{det}(f)$ is incremented by one for every time unit where f is detected by T. A fault f is dropped from further simulation when $n_{det}(f)$ reaches n. This is consistent with the first definition of an n-detection test sequence given in [5]. Using the numbers of detections $n_{det}(f)$, we can define the total and average number of detections, as well as the metric based on the subsets F_m in the same way as for a combinational test set.

3.2. The quality of transparent-scan sequences

Several effects may cause a test sequence T, which is translated from a combinational test set C, to have a higher quality than C as an n-detection test set. (1) Under T, there is no distinction between scan clock cycles (where $s_{sel} = 1$) and functional clock cycles (where $s_{sel} = 0$). The test sequence T is simulated during both types of clock cycles, and fault detections are captured on the original primary outputs during both types of clock cycles. For a combinational test set C, fault detections are captured on the original primary outputs only during functional clock cycles. As a result, T may have higher numbers of fault detections than C. (2) We have the option of specifying the unspecified values of T so as to improve the numbers of detections of faults. It is important to note that the number of clock cycles required for applying C and T (translated from C) are the same. Thus, we obtain improved n-detection quality for the same test application time. Next, we consider each one of the effects listed above.

We first compare C with the transparent-scan sequence T before any unspecified values in T are specified. We consider the example of $s27$. In row *comb* of Table 3 we show the results of 6-detection fault simulation for the combinational test set C of $s27$ from Table 1. We show the average number of detections under column *ave*. We then show the size of F_m under C, for $1 \leq m \leq 6$.

Table 3: Quality of tests for $s27$

type	ave	m=1	m=2	m=3	m=4	m=5	m=6
comb	2.00	15	7	5	5	0	0
x	3.25	5	9	5	4	4	5
rnd	4.28	3	3	5	5	3	13
cpi	4.38	3	2	5	6	2	14

In row x of Table 3 we show the results of 6-detection fault simulation for the transparent-scan test sequence T_x obtained by translating C. This sequence is shown in Table 2 under column T_x. It can be seen that the average number of detections is higher for T_x than for C. More important, C has 15 faults that are detected only once. Under T_x, only five faults are detected once, and the remaining faults are detected higher numbers of times. This is a result of the fact that T_x provides more time units where faults may be detected.

We can specify the unspecified values of T_x randomly to obtain the test sequence T_{rnd} shown in Table 2 under column T_{rnd}. The results of 6-detection fault simulation for T_{rnd} are shown in row *rnd* of Table 3. It can be seen that specification of the unspecified values in T_x improves the quality of the transparent-scan sequence.

Next, we consider a different way of specifying the unspecified values of T_x. We concentrate on the unspecified values of the original primary inputs of the circuit, which are obtained when $s_{sel} = 1$. We will specify the unspecified values on s_{inp} when $s_{sel} = 0$ randomly. We note that each test vector $c_i \in C$ includes a subvector c_{iI} for the original primary inputs of the circuit. This subvector may be useful in detecting faults at any time unit of T_x. We define $C_I = \{c_{iI} : c_i \in C\}$. We use the primary input subvectors in C_I as follows. For every time unit u of T_x where $s_{sel} = 1$ and the original primary inputs are unspecified, we randomly select an index i of a subvector $c_{iI} \in C_I$. We then assign c_{iI} to the primary inputs at time unit u of T_x. We also set $CI\_ind(u) = i$ to indicate that c_{iI} is used at time unit u of the sequence. This will be useful later. We denote the resulting test sequence by T_{cpi} (*cpi* stands for *combinational primary input* vectors).

A test sequence T_{cpi} based on the test sequence T_x of $s27$ is shown under column T_{cpi} of Table 2. Before every vector of T_{cpi} we show the index of the combinational test vector whose primary input subvector is used in T_{cpi}, $CI\_ind(u)$.

We show the results of 6-detection fault simulation of T_{cpi} for $s27$ in row *cpi* of Table 3. It can be seen that the use of primary input vectors based on C improves the quality of the transparent-scan test sequence relative to the use of random values. In the following subsection we describe a procedure for further improving the quality of the test sequence T_{cpi}.

3.3. Modifying T_{cpi} to improve its quality

We first demonstrate the modification procedure of T_{cpi} by considering the test sequence T_{cpi} of $s27$ in Table 2.

Based on Table 1, the candidate primary input vectors for $s27$ are $C_I = \{c_{iI} : 0 \le i \le 5\}$. The time units where primary input vectors from C_I are embedded in T_{cpi} are $U = \{0, 1, 2, 4, 5, 6, \cdots, 24, 25, 26\}$. The index of the primary input vector included in T_{cpi} at time unit u, $CI\_ind(u)$, is shown in Table 2 for $0 \le u \le 26$. We consider the time units in U one at a time in a random order. For every time unit u, we attempt to replace $c_{CI\_ind(u)I}$ with a different subvector c_{iI}. We select the index i randomly. We accept the replacement only if the n-detection quality of T_{cpi} is improved.

When $u = 2$ is considered, we attempt to replace $c_{CI\_ind(2)} = c_{3I} = 0111$ with $c_{1I} = 1001$ ($u = 2$ and $i = 1$ are selected randomly). We find that the number of detections of the fault f_7 will go down from four to three. We therefore keep c_{3I} at time unit $u = 2$.

When $u = 14$ is considered, we attempt to replace $c_{CI\_ind(14)} = c_{5I} = 1010$ with $c_{3I} = 0111$ ($u = 14$ and $i = 3$ are selected randomly). We find that the number of detections of the fault f_6 changes from six to five; the number of detections of f_{14} changes from five to six; the number of detections of f_{19} changes from three to four; the number of detections of f_{21} changes from six to five; and the number of detections of f_{31} changes from five to six. The effect on the metric based on the sets F_m is shown in Table 4. The sizes of the sets F_m for T_{cpi} are shown in row *cpi*, and the sizes of the sets F_m for the modified test sequence are shown in row *mod*. Overall, the quality of the modified test sequence is better, and we accept the change. Note that we accept the change even though the number of detections went down for some faults with higher numbers of detections. The reduction in the number of faults with three detections from five to four determines that the modified test sequence is better.

Table 4: Effect of modifying T_{cpi}

type	m=1	m=2	m=3	m=4	m=5	m=6
cpi	3	2	5	6	2	14
mod	3	2	4	7	2	14

After we consider all the time units in U once we consider them again until two consecutive passes over U do not improve the quality of the test sequence.

To reduce the simulation effort required for computing the quality metric of a modified test sequence, we only consider faults that are detected up to n_0 times by T_{cpi}, for a constant $n_0 < n$. Thus, we compute the sets F_m over a subset of the faults that have $1 \le n_{det}(f) \le n_0$ before the modification, and we use these sets to determine whether a modification of T_{cpi} is acceptable. By using $1 \le m \le n_0$, we ignore the effect of the modification on faults that are detected more than n_0 times. This is justified since their effect on the overall quality is lower. Even if they are detected fewer times after a modification, an increase in the number of detections of faults detected fewer times is advantageous. We accept a modification only if such an improvement occurs.

After we accept a modification, we compute the numbers of detections of all the faults under the modified test sequence to ensure that the next modification is done using accurate information.

The procedure for modifying T_{cpi} is given next as Procedure 1. We denote by T_{mod} the test sequence that results from applying Procedure 1 to T_{cpi}

Procedure 1: Modifying T_{cpi}

(1) Let $C = \{c_0, c_1, \cdots, c_{k-1}\}$ be a combinational test set. Let $T_{cpi} = t_0 t_1 \cdots t_{L-1}$ be the transparent-scan sequence with original primary input vectors out of $C_I = \{c_{0I}, c_{1I}, \cdots, c_{(k-1)I}\}$. If t_u contains a subvector $c_{iI} \in C_I$, $CI\_ind(u) = i$; otherwise, $CI\_ind(u) = -$. Set $T_{mod} = T_{cpi}$.

(2) Perform n-detection fault simulation of T_{mod}. Store for every fault f the number of times $n_{det}(f)$ the fault is detected by T_{mod}.

(3) Include in U every time unit u such that $CI\_ind(u) \ge 0$.

(4) Select a time unit $u \in U$ randomly. Remove u from U.

(5) Select a subvector $c_{iI} \in C_I$ randomly such that $CI\_ind(u) \ne i$.

(6) Replace $c_{CI\_ind(u)I}$ in t_u with c_{iI}.

(7) Let $\hat{F} = \{f : n_{det}(f) \le n_0\}$. Simulate T_{mod} under every fault $f \in \hat{F}$. Considering only the faults in $\hat{F}$, if the quality of T_{mod} is improved by the change in t_u:
   (a) Set $CI\_ind(u) = i$.
   (b) Perform n-detection fault simulation of T_{mod}. Store for every fault f the number of times $n_{det}(f)$ the fault is detected by T_{mod}.
   Otherwise, replace c_{iI} in t_u with $c_{CI\_ind(u)I}$.

(8) If $U \ne \phi$, go to Step 4.

(9) If T_{mod} was not modified in two consecutive passes over all the time units in U, stop.

(10) Go to Step 3.

4. Experimental results

We consider ISCAS-89 and ITC-99 benchmark circuits with $n = 10$. For the combinational test set C we use a compact single-detection test set. We consider four transparent-scan sequences for every circuit. (1) The incompletely-specified test sequence T_x obtained after translation of C. (2) The test sequence T_{rnd} obtained after specifying the unspecified values of T_x randomly. (3) The test sequence T_{cpi} obtained after using original primary input vectors in C to specify unspecified values of T_x. (4) The test sequence T_{mod} obtained by applying Procedure 1 to T_{cpi}. We use $n_0 = 2$ in Procedure 1.

We always simulate C using a combinational fault simulation procedure. We also perform limited simulation of T_x, where fault effects are observed on the original primary outputs only when $s_{sel} = 0$, and on the scan output(s) only when $s_{sel} = 1$. This simulation process imitates combinational fault simulation while eliminating the following effects. (1) There are slight variations in fault lists when scan is inserted into a circuit. (2) Combinational fault simulation counts a fault as detected once even if the fault is propagated to several flip-flops and results in a fault effect at several scan clock cycles.

We consider circuits with 1, 2, 4, $\cdots$ scan chains. We use the smallest number of scan chains that results in a transparent-scan sequence of length 5000 or less. We denote the circuit $circ$ that has S scan chains by $circ.S$.

Table 5: Numbers of detections

circuit	type	ave	m=1	m=2	m=3	m=4	m=5	m=10
s510.1	comb	5.45	113	76	51	44	27	178
	lim	5.70	104	74	46	41	29	199
	x	7.72	47	46	30	16	18	362
s526.1	comb	5.45	92	112	48	30	28	189
	lim	5.68	89	100	45	32	28	204
	x	9.24	0	4	2	5	2	431
s641.1	comb	4.97	99	77	42	25	31	109
	lim	6.04	67	70	31	22	32	180
	x	6.13	67	70	31	22	30	193
	rnd	8.52	20	16	22	7	13	341
	cpi	9.36	3	6	10	8	9	406
	mod	9.36	1	3	14	10	8	405
s820.1	comb	3.54	336	180	75	27	28	136
	lim	3.71	324	166	72	29	43	144
	x	5.07	162	142	82	84	51	235
	rnd	6.00	93	120	76	80	65	314
	cpi	6.02	85	125	80	80	54	309
	mod	6.11	67	116	100	88	47	312
s953.1	comb	5.58	205	159	102	75	37	387
	lim	6.38	135	142	81	81	37	484
	x	9.52	5	16	9	16	5	969
s1196.1	comb	6.11	239	139	72	58	62	522
	x	6.65	141	172	74	58	51	607
	rnd	8.60	25	60	46	37	39	919
	cpi	9.59	8	8	13	13	12	1131
	mod	9.61	0	8	19	14	12	1134
s1423.1	comb	5.36	291	164	165	102	107	388
	x	8.51	8	61	126	34	19	1057
	rnd	8.62	5	47	127	43	11	1096
	cpi	8.66	5	48	104	58	19	1104
	mod	8.66	5	46	106	58	19	1104
s5378.4	comb	7.66	359	356	203	197	167	2837
	x	9.17	72	129	104	76	60	3899
	rnd	9.50	48	74	72	51	26	4182
	cpi	9.52	49	74	73	43	22	4199
s9234.8	comb	6.11	1058	846	501	351	296	2753
	x	7.22	545	672	348	361	250	3592
s15850.16	comb	6.89	1596	1171	648	445	342	6160
	x	8.41	621	570	360	309	246	8320
	rnd	8.47	611	543	360	307	251	8568
	cpi	8.48	607	547	362	293	251	8565
s35932.8	comb	3.27	9025	8227	5070	3531	2536	145
	x	4.66	8522	7121	3791	2066	1046	9120
b04.1	comb	7.67	63	74	77	87	78	770
	x	9.71	0	11	9	5	11	1253
b09.1	comb	5.88	40	48	51	37	37	122
	x	8.10	7	31	17	20	11	262
b10.1	comb	6.04	83	60	45	33	27	207
	x	8.36	19	12	17	35	19	348
b11.1	comb	6.71	119	106	78	75	59	500
	x	9.34	0	12	29	21	20	934
b14.16	comb	7.93	264	359	465	541	413	4982
	x	9.13	129	161	165	226	121	6778

The results are given in Table 5. In row *comb* we show the results of 10-detection fault simulation for the combinational test set C. In row *lim* we show for some of the circuits the results of limited 10-detection fault simulation for the transparent-scan sequence T_x as described above. In rows x, rnd, cpi and mod we show the results of full 10-detection fault simulation for the transparent-scan sequences T_x, T_{rnd}, T_{cpi} and T_{mod}, respectively. In every case we show the average number of detections under column ave, and the size of F_m under column $m = \hat{m}$, for $1 \le \hat{m} \le 5$ and $\hat{m} = 10$. We omit the results for T_{rnd}, T_{cpi} and T_{mod} if the quality of the sequence does not change relative to T_x. The following points can be seen from Table 5.

The quality of T_x as an n-detection test set is always better than the quality of C. This is due to the fact that more input vectors are available for detecting faults under T_x, even though the number of clock cycles for applying T_x is the same as that of C, and T_x is incompletely specified.

Specifying T_x improves the quality of the test sequence in many cases. In the cases where the quality is not improved, fault detection has only a weak dependence on the original primary input vectors. This typically happens when the circuit has a small number of primary inputs. In the cases where the quality is improved, T_{cpi} is in most cases better than T_{rnd} as an n-detection test sequence. Procedure 1 is able to improve the quality of the test sequence even further in these cases.

5. Generating n-detection sequences

The transparent-scan sequences T_x, T_{rnd}, T_{cpi} and T_{mod} detect faults more times than C. However, they do not reach n detections for each target fault, even for small values of $n > 1$. In this section we complete a transparent-scan sequence into an n-detection sequence. We use T_{rnd} as the initial test sequence. Other test sequences can be used in a similar way.

We perform n-detection fault simulation to find the number of times $n_{det}(f)$ every fault f is detected by T_{rnd}. Since C is a single-detection test set, T_{rnd} is also guaranteed to be a single-detection test sequence, i.e., $n_{det}(f) \ge 1$ for every target fault f. For $d = 2,3, \cdots ,n$, we extend T_{rnd} into a d-detection test sequence by targeting every fault f such that $n_{det}(f) < d$. When a fault f is targeted, we attempt to generate a test subsequence $T(f)$ that detects f starting from the final states reached under T_{rnd}. We search for $T(f)$ by performing sequential test generation. If the fault is activated on a flip-flop but not detected, we use an activation sequence followed by a scan-out operation of the appropriate length to detect it. If a test subsequence $T(f)$ that detects f is found, then $T_{rnd}T(f)$ detects f one additional time compared to T_{rnd}. We replace T_{rnd} by $T_{rnd}T(f)$ and perform n-detection fault simulation in order to update the numbers of detections of all the faults. We repeat this process until all the faults are detected d times. We then consider the next value of d.

To reduce the n-detection fault simulation effort during test generation, we only simulate a fault f with $n_{det}(f) < d$ during d-detection test generation, since only such a fault may have to be targeted. We perform a complete n-detection fault simulation pass only after the d-detection test sequence is obtained. In addition, we drop a fault from consideration once it is detected n times.

Table 6: N-detection test generation

circuit		d=1	d=2	d=3	d=4	d=5	d=10
s510.1	len	384	537	639	685	722	869
	rtio	1.00	1.40	1.66	1.78	1.88	2.26
	ndet<d	0	0	0	0	0	0
s526.1	len	1121	1121	1140	1193	1241	2948
	rtio	1.00	1.00	1.02	1.06	1.11	2.63
	ndet<d	0	0	0	0	0	10
s641.1	len	459	768	1255	1764	2168	4288
	rtio	1.00	1.67	2.73	3.84	4.72	9.34
	ndet<d	0	0	0	1	1	1
s820.1	len	569	850	1163	1363	1568	2509
	rtio	1.00	1.49	2.04	2.40	2.76	4.41
	ndet<d	0	7	1	3	5	7
s953.1	len	2309	2327	2394	2445	2518	3193
	rtio	1.00	1.01	1.04	1.06	1.09	1.38
	ndet<d	0	0	0	0	0	0
s1196.1	len	2640	2770	3288	3681	4063	6423
	rtio	1.00	1.05	1.25	1.39	1.54	2.43
	ndet<d	0	4	7	5	5	9
s1423.1	len	2024	2169	2742	3902	4767	8462
	rtio	1.00	1.07	1.35	1.93	2.36	4.18
	ndet<d	0	0	4	8	13	18
b04.1	len	4220	4220	4468	4545	4678	5465
	rtio	1.00	1.00	1.06	1.08	1.11	1.30
	ndet<d	0	0	0	0	0	0
b09.1	len	956	1007	1476	1985	2633	5985
	rtio	1.00	1.05	1.54	2.08	2.75	6.26
	ndet<d	0	0	0	0	0	2
b10.1	len	1043	1400	1626	1792	1962	2801
	rtio	1.00	1.34	1.56	1.72	1.88	2.69
	ndet<d	0	0	0	0	0	0
b11.1	len	2417	2417	2633	2902	3055	4044
	rtio	1.00	1.00	1.09	1.20	1.26	1.67
	ndet<d	0	0	0	0	0	1

Results of this process using $n = 10$ are shown in Table 6. For $1 \le d \le 5$ and for $d = 10$ we show the results of d-detection test generation as follows. In row *len* we show the length L_d of T_{rnd} after d-detection test generation. In row *rtio* we show the test length ratio L_d/L_1, where L_1 is the length of the initial test sequence T_{rnd}. In row *ndet<d* we show the number of faults with $n_{det}(f) < d$. Ideally, this number should be zero if every fault is detected at least d times after d-detection test generation. The following points can be seen from Table 6.

A small number of faults have $n_{det}(f) < d$ at the end of the d-detection test generation process. These faults can be handled by a more complete test generation procedure. Due to the incompleteness of the procedure we use, and due to accidental detections, the number of faults with $n_{det}(f) < d$ may not change monotonically with d.

For many circuits, the growth of the test length with d is slow relative to the approximately linear growth observed for compact combinational test sets. This is due to the fact that additional clock cycles are available for detecting faults compared to a combinational test set. In addition, the sequential test generation procedure is able to interleave scan clock cycles with functional clock cycles in order to derive short test subsequences.

6. Concluding remarks

We studied the quality of test sequences under transparent-scan as n-detection test sequences. The test sequences we considered were obtained from combinational test sets by a process called test translation. A transparent-scan sequence T requires the same number of clock cycles as the combinational test set C from which it is translated, and it is incompletely specified. We showed that under these conditions, a transparent-scan sequence detects faults more times than a combinational test set. This is due to the fact that under transparent-scan, fault detections are captured on all the outputs during scan clock cycles as well as during functional clock cycles.

We considered three procedures for specifying the unspecified values of the transparent-scan sequence. In many cases, the procedure that used randomly selected primary input subvectors out of C increased the numbers of times faults are detected relative to random specification. Other methods of filling unspecified values can be used as well, and are guaranteed to have a quality at least equal to that of the incompletely specified sequence. We also considered the extension of a transparent-scan test sequence into an n-detection test sequence that detects every target fault at least n times.

References

[1] S. C. Ma, P. Franco and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results", in Proc. Intl. Test Conf., 1995, pp. 663-672.

[2] L.-C. Wang, M. R. Mercer and T. W. Williams, "On the Decline of Testing Efficiency as the Fault Coverage Approaches 100%", in Proc. VLSI Test Symp., April 1995, pp. 74-83.

[3] S. M. Reddy, I. Pomeranz and S. Kajihara, "Compact Test Sets for High Defect Coverage", IEEE Trans. on Computer-Aided Design, Aug. 1997, pp. 923-930.

[4] J. T.-Y. Chang, C.-W. Tseng, C.-M. J. Li, M. Purtell and E. J. McCluskey, "Analysis of Pattern-Dependent and Timing-Dependent Failures in an Experimental Test Chip", in Proc. Intl. Test Conf., 1998, pp. 184-193.

[5] I. Pomeranz and S. M. Reddy, "Test Sequences to Achieve High Defect Coverage for Synchronous Sequential Circuits", IEEE Trans. on Computer-Aided Design, Oct. 1998, pp. 1017-1029.

[6] M. R. Grimaila, S. Lee et. al., "REDO - Random Excitation and Deterministic Observation - First Commercial Experiment", in Proc. VLSI Test Symp., 1999, pp. 268-274.

[7] C.-W. Tseng and E. J. McCluskey, "Multiple-Output Propagation Transition Fault Test", in Proc. Intl. Test Conf., 2001, pp. 358-366.

[8] B. Benware, C. Schuermyer, N. Tamarapalli, K.-H. Tsai, S. Ranganathan, R. Madge, J. Rajski and P. Krishnamurthy, "Impact of multiple-detect test patterns on product quality", in Proc. Intl. Test Conf., Sept. 2003, pp. 1031-1040.

[9] S. Venkataraman, S. Sivaraj, E. Amyeen, S. Lee, A Ojha and R. Guo, "An Experimental Stud of n-Detect Scan ATPG Patterns on a Processor", in Proc. 22nd VLSI Test Symp., April 2004, pp. 23-28.

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

[11] J. Grodstein, D. Bhavsar, V. Bettada and R. Davies, "Automatic Generation of Critical-Path Tests for a Partial-Scan Microprocessor", in Proc. Intl. Conf. on Computer Design, Oct. 2003, pp. 180-186.