Definitions of the Numbers of Detections of Target Faults and their Effectiveness in Guiding Test Generation for High Defect Coverage⁺

Irith Pomeranz School of Electrical & Computer Eng. Purdue University W. Lafayette, IN 47907, U.S.A. and

Sudhakar M. Reddy Electrical & Computer Eng. Dept. University of Iowa Iowa City, IA 52242, U.S.A.

Abstract

The number of times a fault f in a combinational circuit is detected by a given test set T was shown earlier to affect the defect coverage of the test set. The earlier definition counted each test in T, that detects f, as a distinct detection of f. This definition counts two tests as distinct detections even if they differ only in the values of inputs that do not affect the activation or propagation of the fault. In this work, we introduce a stricter definition that requires that two counted tests would be different in the way they activate and/or propagate the fault. We describe procedures for constructing test sets based on the stricter definition. The results show a simple criterion to decide when it may be necessary to combine the two definitions in order to obtain a high quality test set.

1. Introduction

n-detection test sets were shown to achieve improved defect coverage in [1]-[6]. An *n*-detection test set is one where each modeled fault is detected either by *n* different tests, or by the maximum number of different tests that can detect the fault if this number is smaller than *n*. *n*-detection test sets for stuck-at faults in combinational circuits were considered in [1]-[4]. Stuck-at faults in sequential circuits were considered in [5], and transition faults in combinational and full-scan circuits were considered in [6].

In the experiments described in [1] and [3], chips were fabricated for the purpose of comparing various test sets and test application processes, including *n*-detection stuck-at test sets. The experiments of [1] and [3] showed that the defect coverage achieved by *n*-detection stuck-at test sets for large enough values of *n*, when applied at-speed, led to the identification of all or most of the defective chips. The experiments described in [2], [5] and [6] demonstrated the effectiveness of *n*-detection test sets by simulating unmodeled faults [7], [8].

The advantage of using *n*-detection test sets over other approaches to generate test sets with high defect coverage is that an existing test generation procedure for a simple fault model (such as stuck-at faults) can be used with simple extensions, thus avoiding the need to develop new test generation procedures for new fault or defect models.

Considering combinational circuits, the existing definition says that the number of times a fault is detected by a test set T is equal to the number of tests in T that detect the fault. This definition allows the following situation. Consider the circuit

illustrated in Figure 1, where the fault f can be detected either on primary output O_1 or on O_2 . Suppose that the input cones of O_1 and O_2 include primary inputs I_1 , I_2 and I_3 . In this case, the values of I_4 and I_5 do not affect the detection of f. Let $t_1 = 00000$ and $t_2 = 00011$ be two tests that detect f. If t_1 and t_2 are included in a test set T, then, according to the existing definition, f is detected twice by T. However, it can be seen that t_1 and t_2 differ only in the values of I_4 and I_5 that do not affect the detection of f. Consequently, a stricter definition of the number of detections of f may be necessary in order to count only one of t_1 and t_2 as a distinct detection of f.



Figure 1: Example circuit 1

In this work, we develop a stricter definition of the number of detections of a fault. Under this definition, only one of t_1 and t_2 above would be counted as a distinct detection of f. We incorporate the proposed definition into a test generation procedure based on test selection, and compare the resulting test sets to the test sets selected based on the earlier, less strict definition. The comparison includes the numbers of detections by both definitions, the test set size, and the coverage of bridging faults which are used as surrogate faults representing unmodeled defects. Two types of results are obtained that support the use of the stricter definition.

For some circuits, the stricter definition results in a smaller test set than the less strict definition. When the test sets are simulated under either definition, the test set based on the stricter definition detects each fault at least as many times as the test set based on the less strict definition. The bridging fault coverage is also larger than that of the test set based on the less strict definition. For such circuits, test generation based on the less strict definition is preferable to test generation based on the less strict definition.

For other circuits, the stricter definition excludes too many tests from the test set because they do not contribute distinct detections of any fault. As a result, the numbers of detections are lower than the target, n. In addition, the numbers of detections and the test set size remain approximately the same even when the target number of detections n is increased. This results in smaller test sets with lower coverage of bridging faults.

⁺ Research supported in part by NSF Grant No. MIP-9725053, and in part by SRC Grant No. 98-TJ-645.

For such circuits, we extend the test set based on the stricter definition by adding tests so as to achieve the target number of detections based on the less strict definition. When this is done, the bridging fault coverage catches up and even exceeds the coverage achieved by the test set based on the earlier definition. We conclude that for such circuits, the stricter definition is effective in guiding the initial phase of test generation, and test generation should then continue using the less strict definition.

Although the stricter definition is developed for stuck-at faults in combinational circuits, it can be extended to other fault models and to sequential circuits.

The paper is organized as follows. In Section 2, we provide the proposed stricter definition of the number of detections of a target fault by a given test set T. In Section 3, we describe procedures for selecting test sets based on the stricter definition. We also describe a similar procedure for the less strict definition. In Section 4, we present experimental results comparing the various test sets. Section 5 concludes the paper.

2. Definitions of the number of detections

The definition used earlier for the number of times a fault f is detected by a test set T counts every test that detects f as a different detection of f. This definition is given as Definition 1 next. **Definition 1:** Let T be a test set where no test is duplicated. A fault f is detected n times by T if T contains n tests that detect f.

Motivated by the example of Section 1, we describe next a stricter definition that requires the tests counted as detecting a fault *f* to be *sufficiently different*.

Consider two tests, $t_1 = 0101$ and $t_2 = 0110$. Suppose that t_1 is the first test in the test set, and that t_2 is the second test in the test set. Suppose that t_1 and t_2 detect a fault f. After simulating t_1 and finding that it detects f, we count t_1 as the first detection of f. According to Definition 1, we count t_2 as a second detection of f when t_2 is simulated. To verify that t_2 is sufficiently different from t_1 before we count it as a second detection of f, let us define a test t_{12} as follows. For inputs where t_1 and t_2 assume the same value, t_{12} assumes the same value as t_1 and t_2 . For inputs where t_1 and t_2 assume different values, t_{12} is left unspecified. For t_1 and t_2 above, we obtain $t_{12} = 01xx$. If t_{12} detects f, this implies that the inputs where t_1 and t_2 differ are not critical to the detection of f. We conclude that the two tests are essentially the same, and we do not count t_2 as a second detection of f. If, on the other hand, t_{12} does not detect f, we conclude that t_1 and t_2 are different on inputs that are important for the detection of f, and we count t_2 as a second detection of f.

Applying these considerations to the fault of Figure 1 with the tests $t_1 = 00000$ and $t_2 = 00011$, we consider $t_{12} = 000xx$. Since t_{12} detects f, we count t_1 but not t_2 as a new detection of f. Definition 2 given next extends this way of counting detections to the case where a test set T includes several tests that detect a fault f. In this case, we require that a test counted as a distinct detection of f would be sufficiently different (by the definition above) from every earlier counted test. For Definition 2, we denote the value of input k under test t by t(k). **Definition 2:** Let T be a test set where no test is duplicated. A fault f is detected n times by T if there exist n tests t_1, t_2, \dots, t_n in T such that the following condition is satisfied. For every i and j such that $1 < i \le n$ and $1 \le j < i$, let t_{ij} be a test where $t_{ij}(k) = t_i(k)$ if $t_i(k) = t_j(k)$, and $t_{ij}(k) = x$ otherwise. The test t_{ij} does not detect f.

We implement the count of the number of detections according to Definition 2 by considering the tests in the order they appear in the test set T, i.e., we do not reorder the tests in order to maximize the value of n. Although the order of the tests in T may not yield the highest count of the number of detections for every fault f, it provides a uniform way of computing the numbers of detections of all the faults without performing additional fault-specific computations.

In an *n*-detection test set according to Definition 1 or 2, every fault f should be detected n times by the appropriate definition. If there do not exist n different tests for a fault f, then an *n*-detection test set should detect f the maximum possible number of times according to the definition considered.

To demonstrate what may appear to be a limitation of Definition 2, we consider next the circuit of Figure 2. Consider three tests for the fault *A* stuck-at 1, $t_1 = 001$, $t_2 = 010$ and $t_3 = 011$. All the tests result in g = 1, and the fault is activated and propagated in essentially the same way. Thus, one may consider a strict definition where only one of these tests would be counted. Assuming that these tests are contained in a test set *T* in this order, let us count the number of detections of *A* stuck-at 1 by Definition 2. We first count t_1 . To check whether t_2 should be counted, we simulate $t_{12}=0xx$. Since t_{12} does not detect the fault, we count t_2 as a second detection. To check whether t_3 should be counted, we simulate $t_{13} = 0x1$. Since this test detects the fault, t_3 is not counted as a third detection. We obtain a total of two detections for the fault *A* stuck-at 1. The reasons for using Definition 2 instead of an even stricter definition are as follows.



Figure 2: Example circuit 2

Definition 2 already excludes from counting some tests that do not contribute to sufficiently different detections. The tests not counted may end up being excluded from a test set if they do not contribute to the detection of any fault. In the circuit of Figure 2, if a bridging fault affects A and B (or A and C), the two tests counted by Definition 2 will detect these faults. Thus, both tests are useful. However, a stricter definition than Definition 2 may exclude one of them. If too many tests are excluded from the test set, the test set becomes too small, and is not likely to achieve high defect coverages. This effect happens already to some extent with Definition 2, as demonstrated later. A stricter definition would lose even more tests, and would defeat the purpose of obtaining test sets having high defect coverages.

3. Selecting *n***-detection test sets**

In this section, we describe the construction of *n*-detection test sets based on Definition 1 and based on Definition 2. The test sets are selected out of a given set of candidate tests T_{cand} . For circuits with small numbers of inputs, T_{cand} is the set of all the input combinations of the circuit. This allows us to select complete *n*-detection test sets for both definitions. For circuits with large numbers of inputs, we use T_{cand} which is an n_0 -detection test set for n_0 larger than our target number of detections *n*. The test set selection process for both definitions proceeds as follows. We first apply fault simulation to find tests in T_{cand} that detect every fault f. We simulate a fault f without fault dropping until we find N tests that detect it, where N is a constant larger than the target number of detections n (N = 10 in our implementation). We denote the set of tests that detect f by T(f). It is important to note that a larger value of N will allow us to work with a larger set T(f), and will increase the flexibility in reaching n detections for every fault. However, it will also increase the simulation effort required.

For Definition 2, we reconsider all the tests in T(f) in the order by which they were added to T(f) (which is also the order they appear in T_{cand}). Let $T(f) = \{t_1, t_2, \dots, t_k\}$. We associate with every test t_i , $1 \le i \le k$, a variable $def 2(f, t_i)$. We set $def 2(f, t_i) = 1$ if t_{ji} does not detect f for every $1 \le j < i$ such that $def 2(f, t_i) = 1$, and we set $def 2(f, t_i) = 0$ otherwise. Thus, $def 2(f, t_i)$ indicates whether or not t_i can be counted as a detection of f based on Definition 2 given the tests in the set $\{t_1, t_2, \dots, t_{i-1}\}$ that may be counted.

To select an *n*-detection test set $T_{n,def 1}$ based on Definition 1, we consider the tests in T_{cand} in the order they appear in T_{cand} . A test is selected if it increases the number of detections n(f) for any fault f which is not yet detected n times. We refer to the selection procedure as Procedure 1.

The procedure for selecting an *n*-detection test set $T_{n,def 2}$ based on Definition 2 is similar to Procedure 1, except that a test is added to $T_{n,def 2}$ if it increases the number of detections of a fault f with n(f) < n based on Definition 2. We use the variables def $2(f,t_i)$ defined above to determine whether a test t_i increases the number of detections of a fault f. Note that we do not recompute the variables def $2(f,t_i)$ based on the selected tests. This does not limit the accuracy of the procedure, for the following reason. Consider a fault f with $T(f) = \{t_{i_1}, t_{i_2}, \dots, t_{i_k}\}$. When we consider the tests for inclusion in $T_{n,def 2}$, we encounter t_{i_1} first, then t_{i_2} , and so on, until t_{i_k} is encountered. As long as n(f) < n, we include in $T_{n,def 2}$ the next test t_{i_i} with def $2(f, t_{i_i}) = 1$. When we consider t_{i_i} , if def $2(f, t_{i_i}) = 1$, we can increment n(f) by one correctly based on the fact that all the tests t_{i_m} such that $i_m < i_i$ and def $2(f, t_{i_m}) = 1$ are already included in $T_{n,det2}$, and the fact that $t_{i_m i_i}$ does not detect f for any such test $t_{i_{w}}$ (otherwise, we would have had def $2(f, t_{i_{j}}) = 0$). If we include in $T_{n,det2}$ a test $t_p \in T(f)$ such that $def 2(f,t_p) = 0$ and $p < i_j$, we can ignore t_p when counting the number of detections of f, and consider only the tests with def $2(f, t_{i_m}) = 1$. The procedure that selects an *n*-detection test set based on Definition 2 is referred to as Procedure 2.

We found experimentally that for some circuits, Definition 2 excludes too many tests as distinct detections for any fault. As a result, the test sets produced by Procedure 2 may be small compared to the test sets produced by Procedure 1. In these cases, the test set size and the number of detections by Definition 2 tend to saturate as n is increased. We conclude that for such circuits, Definition 2 is too strict, and may exclude tests that may detect new defects. To compensate for this effect, we take the following approach. We generate an n-detection test set $T_{n,def 2}$ by Definition 2. We then complete it into an *n*-detection test set T_n by Definition 1. Procedure 3 is used for this purpose. Using Procedure 3, tests are selected based on the stricter Definition 2 as much as possible. The numbers of detections for the selected tests are then recomputed using Definition 1 instead of Definition 2. Then, to ensure a sufficient number of detections of every fault, Definition 1 is used for selecting additional tests.

4. Experimental results

In this section, we describe the results of Procedures 1, 2 and 3.

We capture the following parameters for every test set. The average number of detections by each definition is computed as follows. We simulate each fault only until its number of detections by Definition 1 reaches *N*, where *N* is a preselected constant (N = 10 in our experiments). For Definition 1, the number of tests that detect a fault is also its number of detections. For Definition 2, we reconsider the tests and compute the number of sufficiently different tests as described in Section 2. We denote the number of times a fault *f* is detected according to the appropriate definition by n(f). The average number of detections for the definition being considered is computed as $\overline{n} = \sum_{f \in F} n(f)$.

To evaluate the defect coverage, we use bridging faults as surrogates for defects, as was done in [7], [8]. Both AND and OR type bridging faults are considered for every pair of lines g_1 , g_2 that satisfies the following conditions. (1) Both g_1 and g_2 are outputs of multi-input gates. (2) There is no directed path from g_1 to g_2 or from g_2 to g_1 . (3) g_1 and g_2 are not inputs of the same gate. (4) If the number of circuit lines exceeds 5000, then a pair of lines g_1 and g_2 is considered with probability 0.01. If the number of circuit lines exceeds 10000, then a pair of lines g_1 and g_2 is considered with probability 0.005. The probability is divided by two with every additional 5000 lines. This reduces the number of bridging faults that need to be simulated.

The circuits we consider are shown in Table 1. After the circuit name, we show the number of primary inputs of the circuit, the size of the set of candidate tests T_{cand} from which we select *n*-detection test sets, and the number of bridging faults we consider.

The results of Procedures 1, 2 and 3 for circuits with small numbers of inputs are shown in Tables 2 and 3. Table 2 contains results for a few of the circuits considered. Table 3 contains a summary of the results obtained for all the circuits considered, as explained below. For the circuits considered in Tables 2 and 3, the set of tests T_{cand} from which *n*-detection test sets are selected consists of all the primary input combinations of the circuit. Table 2 is organized as follows. After the circuit name, we show the value of *n*. For each one of the three procedures, we then show the number of tests in the *n*-detection test set produced by the corresponding procedure, the average number of detections by Definition 1, the average number of detections by Definition 2, and the bridging fault coverage. The following points can be seen from Table 2.

(1) Considering Definition 1, Procedure 1 based on Definition 1 results in higher average numbers of detections than Procedure 2 based on Definition 2. This is because Procedure 2 does not accept some of the tests accepted by Definition 1 as increasing the numbers of detections for certain faults. Procedure 3 brings the average number of detections according to Definition 1 closer to that of Procedure 1.

(2) Considering Definition 2, Procedure 2 based on Definition 2 results in higher average numbers of detections than Procedure 1 based on Definition 1. We know that Procedure 3 does not reduce the numbers of detections by Definition 2, since it only adds tests to the test set. However, the averages shown in Table 2 are computed based on the selected test set, without trying to order the tests so as to maximize the numbers of detections of every fault. Consequently, simulation may show a reduction in the average number of detections.

Table 1	: Circuit	parameters

circuit	inp	cand	bridg
Z9sym	9	512	30574
add6	12	4096	8912
adr4	8	256	3882
alu1	12	4096	1530
alu2	10	1024	4992
alu3	10	1024	7740
co14	14	16384	3988
dk17	10	1024	3632
dk27	8	256	710
dk48	15	32768	3568
radd	8	256	2600
rd53	5	32	2692
z4	7	128	2218
s298	17	231	16612
s344	24	134	32210
s382	24	251	29622
s400	24	241	30932
s510	25	413	50190
s526	24	493	43330
s641	54	212	144752
s820	23	844	90696
s953	45	771	175568
s1196	32	1112	283014
s1423	91	233	463760
s5378	214	989	87908
s9234	247	1091	334680
s13207	700	2333	372244
s15850	611	964	351554
s35932	1763	192	471056

(3) The average number of detections by Definition 2 tends to saturate as n is increased, even when Procedure 2 is used. This indicates that Procedure 2 does not have enough candidate tests to improve the numbers of detections by Definition 2. This also affects the test set size and the bridging fault coverage discussed next.

(4) The test set sizes produced by Procedure 1 tend to be the highest of all three procedures. Procedure 2 selects relatively small numbers of tests because it cannot increase the numbers of detections by Definition 2 any further. In addition, the test set sizes produced by Procedure 2 tend to saturate as n is increased. Procedure 3 adds some tests to the test set produced by Procedure 2, but generally, the total number of tests required to achieve an n-detection test set by Definition 2 and then complement it based on Definition 1 is smaller than if only Definition 1 is used.

(5) The bridging fault coverage shows an advantage to the test set produced by Procedure 3 over the ones produced by Procedures 1 and 2.

The last two points can be seen more clearly from Table 3. In Table 3, we include for every value of n and every procedure the average test set size and the average bridging fault coverage. The average is computed over all the circuits using the corresponding procedure and value of n.

The results of Procedures 1, 2 and 3 for several circuits with larger numbers of inputs are shown in Table 4. Averages for all the circuits considered are shown in Table 5. For these circuits, the sets of tests T_{cand} , from which *n*-detection test sets are selected, are 10-detection test sets computed in [2] based on

Table 2: Circuits with small numbers of inputs

			Proce	edure	1		Proc	edure	2		Proce	edure	3
circuit	n	tst	def1	def2	bridg	tst	def1	def2	bridg	tst	def1	def2	bridg
add6	1	55	5.92	2.32	98.11	55	5.92	2.32	98.11	55	5.92	2.32	98.11
	2	101	7.66	2.10	98.49	68	6.70	2.53	98.50	92	7.60	2.42	98.61
	3	145	8.49	1.89	98.78	80	7.27	2.62	98.50	133	8.48	2.12	98.82
	4	184	9.03	1.74	98.79	80	7.27	2.62	98.50	176	9.00	1.86	98.82
adr4	1	38	6.12	2.80	97.81	38	6.12	2.80	97.81	38	6.12	2.80	97.81
	2	67	7.75	2.60	98.20	52	7.03	2.93	98.12	54	7.19	2.97	98.12
	3	87	8.51	2.60	98.20	61	7.55	3.09	98.17	71	8.06	3.06	98.22
	4	106	9.09	2.54	98.22	68	7.94	3.20	98.22	95	8.90	2.85	98.25
alu1	1	30	4.04	1.66	97.45	30	4.04	1.66	97.45	30	4.04	1.66	97.45
	2	68	6.80	1.63	98.89	36	4.94	1.68	98.30	58	6.50	1.69	98.89
	3	96	7.57	1.60	99.15	36	4.94	1.68	98.30	90	7.43	1.61	99.15
	4	132	8.19	1.51	99.15	36	4.94	1.68	98.30	126	8.14	1.51	99.15
alu2	1	60	6.17	2.23	99.44	60	6.17	2.23	99.44	60	6.17	2.23	99.44
	2	112	7.60	1.78	99.44	82	7.19	2.54	99.50	98	7.60	2.37	99.50
	3	160	8.31	1.68	99.60	87	7.38	2.58	99.62	131	8.14	2.21	99.68
	4	210	8.73	1.61	99.68	91	7.53	2.62	99.62	177	8.62	1.92	99.68
dk17	1	28	5.55	2.92	99.34	28	5.55	2.92	99.34	28	5.55	2.92	99.34
	2	52	7.01	2.72	99.70	43	6.61	3.03	99.67	52	7.07	2.97	99.67
	3	74	7.66	2.68	99.81	52	6.88	2.94	99.78	76	7.69	2.78	99.78
	4	100	8.10	2.62	99.81	54	6.94	2.90	99.81	96	8.05	2.73	99.81
dk48	1	29	5.72	3.19	99.89	29	5.72	3.19	99.89	29	5.72	3.19	99.89
	2	60	7.03	2.76	99.89	47	6.54	3.06	99.92	61	7.09	2.89	99.92
	3	91	7.72	2.57	99.92	50	6.66	3.01	99.92	87	7.69	2.70	99.92
	4	129	8.15	2.47	99.92	53	6.76	2.96	99.92	118	8.09	2.54	99.92
radd	1	28	5.70	2.53	98.00	28	5.70	2.53	98.00	28	5.70	2.53	98.00
	2	48	7.75	2.40	98.23	41	7.14	2.90	98.27	44	7.41	2.89	98.35
	3	70	8.72	2.26	98.38	47	7.84	3.03	98.46	57	8.30	2.93	98.46
	4	86	9.16	2.08	98.42	55	8.33	3.16	98.46	77	9.10	2.79	98.54

Table 3: Circuits with small numbers of inputs (averages)

	Procee	lure 1	Proce	dure 2	Proced	lure 3
n	tests	bridg	tests	bridg	tests	bridg
1	46.85	98.02	46.85	98.02	46.85	98.02
2	81.23	98.33	67.23	98.29	76.00	98.36
3	108.31	98.42	74.31	98.33	99.38	98.44
4	134.62	98.44	78.85	98.34	125.23	98.45

Definition 1. The following points can be seen from Tables 4 and 5.

(1) Some of the circuits in Table 4, including s 298 and s 13207, behave similar to the circuits of Table 2, i.e., Procedure 2 selects smaller test sets than Procedure 1, the number of detections by Definition 2 and the test set size tend to saturate as n is increased, and the test sets result in lower bridging fault coverages. Procedure 3 is required for these circuits in order to increase the numbers of detections by Definition 1. This increases the test set size, but the bridging fault coverage is equal or higher in most cases.

(2) For other circuits in Table 4, Procedure 2 is sufficient to generate test sets with high numbers of detections by both Definition 1 and 2. The resulting bridging fault coverages are equal to or higher than those achieved by the test sets produced by Procedure 1. Procedure 3 is not needed in these cases to complement the *n*-detection test sets produced based on Definition 2.

Table 4:	Circuits	with	larger	num	bers	of inpu	ts
----------	----------	------	--------	-----	------	---------	----

		Procedure 1			Procedure 2			Procedure 3					
circuit	n	tst	def1	def2	bridg	tst	def1	def2	bridg	tst	def1	def2	bridg
s298	1	33	5.60	1.98	99.28	33	5.60	1.98	99.28	33	5.60	1.98	99.28
	2	61	7.23	1.91	99.37	40	6.04	2.01	99.30	59	7.16	1.98	99.37
	3	87	8.01	1.81	99.41	42	6.16	2.02	99.31	82	7.92	1.94	99.41
	4	111	8.60	1.77	99.46	44	6.27	2.02	99.31	108	8.57	1.85	99.46
s510	1	77	6.28	2.93	98.30	77	6.28	2.93	98.30	77	6.28	2.93	98.30
	2	143	7.69	2.83	98.63	96	6.80	2.97	98.42	128	7.48	2.93	98.52
	3	196	8.35	2.74	98.76	103	6.97	2.96	98.47	186	8.26	2.84	98.77
	4	250	8.78	2.70	98.84	107	7.04	2.94	98.47	240	8.73	2.77	98.85
s526	1	54	5.72	2.24	99.19	54	5.72	2.24	99.19	54	5.72	2.24	99.19
	2	108	7.39	2.37	99.36	105	7.25	2.46	99.35	112	7.45	2.46	99.37
	3	162	8.35	2.41	99.38	114	7.44	2.48	99.37	158	8.27	2.46	99.41
	4	217	8.95	2.42	99.41	118	7.52	2.50	99.37	208	8.88	2.46	99.42
s1196	1	174	6.76	3.23	98.29	174	6.76	3.23	98.29	174	6.76	3.23	98.29
	2	316	7.87	3.28	98.63	280	7.63	3.60	98.71	306	7.85	3.58	98.73
	3	439	8.43	3.29	98.63	340	7.96	3.73	98.77	417	8.42	3.67	98.83
	4	556	8.81	3.34	98.81	368	8.07	3.79	98.80	516	8.76	3.67	98.87
s1423	1	51	7.52	3.89	99.44	51	7.52	3.89	99.44	51	7.52	3.89	99.44
	2	80	8.58	4.36	99.54	94	8.89	4.55	99.56	94	8.89	4.55	99.56
	3	107	9.06	4.63	99.57	129	9.36	4.80	99.58	129	9.36	4.80	99.58
	4	134	9.40	4.82	99.58	159	9.61	4.92	99.59	159	9.61	4.92	99.59
s5378	1	131	8.40	4.87	99.80	131	8.40	4.87	99.80	131	8.40	4.87	99.80
	2	217	8.93	5.08	99.89	262	9.12	5.17	99.89	266	9.14	5.17	99.89
	3	327	9.29	5.20	99.91	394	9.45	5.29	99.91	398	9.46	5.29	99.91
	4	429	9.50	5.28	99.92	500	9.59	5.34	99.92	507	9.60	5.34	99.92
s9234	1	180	7.19	4.02	98.64	180	7.19	4.02	98.64	180	7.19	4.02	98.64
	2	284	7.92	4.37	98.90	319	8.10	4.45	98.92	321	8.11	4.45	98.93
	3	396	8.36	4.54	99.01	438	8.51	4.62	99.02	444	8.53	4.62	99.03
	4	510	8.67	4.67	99.05	554	8.77	4.74	99.08	566	8.80	4.74	99.08
s13207	1	244	8.38	2.83	99.59	244	8.38	2.83	99.59	244	8.38	2.83	99.59
	2	484	8.79	2.98	99.67	480	8.80	3.05	99.67	494	8.81	3.05	99.67
	3	722	8.98	3.01	99.68	580	8.87	3.06	99.67	723	8.99	3.04	99.68
	4	951	9.13	3.02	99.69	625	8.89	3.06	99.67	953	9.13	3.04	99.69
s15850	1	175	8.33	3.45	99.66	175	8.33	3.45	99.66	175	8.33	3.45	99.66
	2	249	8.73	3.60	99.74	325	9.02	3.70	99.78	328	9.02	3.70	99.78
	3	338	9.04	3.67	99.77	413	9.23	3.74	99.80	417	9.23	3.74	99.80
	4	415	9.23	3.71	99.79	477	9.34	3.75	99.80	490	9.36	3.75	99.80
s35932	1	94	8.32	3.52	98.90	94	8.32	3.52	98.90	94	8.32	3.52	98.90
	2	125	8.66	3.61	98.95	141	8.83	3.63	98.96	142	8.83	3.63	98.96
	3	156	8.83	3.64	98.97	166	8.88	3.65	98.97	168	8.88	3.65	98.97
	4	173	8.88	3.65	98.97	181	8.89	3.65	98.97	183	8.89	3.65	98.97

Table 5: Circu	uits with	ı large	numbers	of inputs
	(av	erages)	

	Procee	lure 1	Proced	lure 2	Procedure 3		
n	tests	bridg	tests	bridg	tests	bridg	
1	96.75	98.87	96.75	98.87	96.75	98.87	
2	168.56	99.12	165.69	99.10	179.50	99.15	
3	239.06	99.17	208.75	99.14	250.44	99.21	
4	305.94	99.22	238.75	99.16	316.69	99.24	

The two types of circuits can be distinguished by considering the test set size produced by Procedure 2 as n is increased. If the test set size saturates, Procedure 3 should be used to complement the test set.

We point out that the results of Tables 2 and 3 were obtained using a selection procedure that has all the input combinations available to it, while the test sets of Tables 4 and 5 were selected out of a limited set of patterns. It is expected that the results of Tables 4 and 5 can be improved if direct test generation is used to produce the same numbers of detections according to Definition 2 (possibly complemented by Definition 1).

5. Concluding remarks

We introduced a new definition of the number of times a fault f is detected by a given test set T. The new definition requires that two counted tests would be sufficiently different. Two tests t_1 and t_2 that detect a fault f are considered sufficiently different if the test t_{12} , which is identical to t_1 and t_2 when they are equal and unspecified otherwise, does not detect f. In this case, t_1 and t_2 differ in values that are necessary for the detection of f, and are thus considered sufficiently different. We described procedures for constructing test sets based on the new definition. The first procedure considered only the new, stricter definition. For cases where this definition excluded too many tests and resulted in low numbers of detections, the second procedure continued to add tests so as to reach the target number of detections by the earlier, less strict definition. For comparison, we also generated test sets based on the earlier definition. Comparison of the three types of test sets showed that the new definition is advantageous either on its own when it produces large enough test sets (or when the test set size does not saturate as *n* is increased), or when complemented by using the earlier definition to select additional tests.

References

- S. C. Ma, P. Franco and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results", in Proc. 1995 Intl. Test Conf., Oct. 1995, pp. 663-672.
- [2] 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.
- [3] 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. 1998 Intl. Test Conf., Oct. 1998, pp. 184-193.
- [4] M. R. Grimaila, S. Lee et. al., "REDO Random Excitation and Deterministic Observation - First Commercial Experiment", in Proc. 17th VLSI Test Symp., April 1999, pp. 268-274.
- [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] I. Pomeranz and S. M. Reddy, "On *n*-Detection Test Sets and Variable *n*-Detection Test Sets for Transition Faults", in Proc. 17th VLSI Test Symp., April 1999, pp. 173-179.
- [7] K. M. Butler and M. R. Mercer, "Quantifying Non-Target Defect Detection by Target Fault Test Sets", in Proc. 1991 Europ. Test Conf., April 1991, pp. 91-100.
- [8] 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.