Generation of Broadside Transition Fault Test Sets that Detect Four-Way Bridging Faults

Irith Pomeranz[1]
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907, U.S.A.

and

Sudhakar M. Reddy[2]
Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242, U.S.A

Abstract

Generation of n-detection test sets is typically done for a single fault model. In this work we investigate the generation of n-detection test sets by pairing each fault of a target fault model with n faults of a different fault model. Tests are generated such that they detect both faults of a pair. To facilitate test generation, we ensure that the faults included in a single pair have overlapping requirements for their detection. The advantage of this approach is that it ensures the detection of additional faults that would not be targeted during n-detection test generation for a single fault model. Experimental results with transition faults as the first fault model and four-way bridging faults as the second fault model are presented.

1. Introduction

Test generation procedures that detect target faults multiple times, resulting in n-detection test sets, were shown to improve the coverage of unmodeled defects [1]-[11]. The basic definition of an n-detection test set requires that each target fault would be detected by n different tests. Other definitions incorporate additional constraints that are aimed at ensuring that different tests for the same target fault will be different in ways that are likely to maximize the coverage of untargeted faults and defects.

All the definitions of n-detection test sets [1]-[11] are based on a single target fault model, typically stuck-at faults. Transition faults have also been considered as targets for n-detection test generation [5]. In this work, we propose a new method that utilizes the extra tests included in an n-detection test set to target a second fault model. Let F_1 be a set of faults of one model, M_1, for which n-detection test generation is to be carried out. Let F_2 be a set of faults of a model M_2 different from M_1. To generate an n-detection test set for F_1, we define n fault pairs based on every fault $f_1 \in F_1$. The first fault of a pair is f_1. The second fault is selected out of F_2. A test t is said to detect the fault pair (f_1,f_2) if t detects f_1 when it is present alone in the circuit, *and* t detects f_2 when it is present alone in the circuit. Test generation is carried out so as to detect every selected fault pair whose faults are detectable by the same test.

The advantage of targeting a second fault model is that it ensures the detection of additional faults, and the defects associated with them, which would not be targeted by an n-detection test set for a single fault model.

We note that different fault models may have overlapping requirements for fault detection of certain faults. It is thus possible to simplify the test generation process for fault pairs by selecting every pair (f_1,f_2) such that a test for f_1 satisfies most of the requirements necessary for the detection of f_2. When f_1 and f_2 have common detection requirements, it is possible to obtain a test for (f_1,f_2) by modifying a test for f_1 such that it would also detect f_2, without losing the detection of f_1. We define fault pairs with common detection requirements, and generate tests by modifying tests for F_1. We start the test generation process from a single-detection test set for F_1.

We demonstrate the proposed process by considering transition faults [12] as the first model, and four-way bridging faults as the second model [13], [14]. We consider broadside tests for transition faults [15], and we detect four-way bridging faults on the second pattern of the test. It is possible to use other fault models, and different types of tests for delay faults. It is also possible to allow bridging faults to be detected on the primary outputs by the first pattern of the test.

The paper is organized as follows. In Section 2 we describe the selection of target fault pairs where the first fault is a transition fault and the second fault is a four-way bridging fault. In Section 3 we describe a test generation procedure for fault pairs. Experimental results of test generation are given in Section 4.

2. Target fault pairs

In this section we describe the selection of target fault pairs. Fault pairs are selected such that the faults share most of the conditions necessary for their detection. Other methods of selecting fault pairs can be used.

Let f_1 be the $a \rightarrow a'$ transition fault on a line g, where $a \in \{0,1\}$. A test for f_1 is a two-pattern test $<u,v>$ such that u sets $g = a$, and v detects the fault g stuck-at a [16].

1. Research supported in part by SRC Grant No. 2004-TJ-1244.
2. Research supported in part by SRC Grant No. 2004-TJ-1243.

Let f_2 be a four-way bridging fault represented as (h_1=b,h_2=b'). The fault f_2 is activated on h_1 when $h_1 = b$ and $h_2 = b'$. A test for f_2 is a single pattern w that detects the fault h_1 stuck-at b' while setting $h_2 = b'$.

Let us consider the pair of faults (f_1,f_2), where f_1 is the $a \rightarrow a'$ transition fault on g, and f_2 is the bridging fault (h_1=b,h_2=b'). If we select f_1 and f_2 such that $h_1 = g$ and $b = a'$, both the second pattern v of a test for f_1 and the single pattern test w for f_2 will have to detect the fault g stuck-at a (which is the same as h_1 stuck-at b'). Thus, a two-pattern test $<u,v>$ that detects both faults will have to satisfy the following conditions (both faults are detected after the second pattern is applied).

(1) u sets $g = a$.

(2) v detects the fault g stuck-at a.

(3) v sets $h_2 = a$.

For uniformity and in order to increase the potential of detecting untargeted delay faults and defects, we add the following fourth condition.

(4) u sets $h_2 = a'$.

After adding this condition we can represent a pair of faults (f_1,f_2) with a common requirement to detect the fault g stuck-at a using the notation (g=$a \rightarrow a'$,h=$a' \rightarrow a$). The first component, g=$a \rightarrow a'$, stands for the transition fault f_1. The second component, h=$a' \rightarrow a$, stands for the requirement to set $h = a'$ under the first pattern, and $h = a$ under the second pattern of the test. Together, they ensure the detection of the $a \rightarrow a'$ transition fault on g, and of the bridging fault (g=a',h=a), with the additional requirement that $h = a'$ under the first pattern of the test.

For illustration, we show in Figure 1 a two-pattern test for the fault (g=1$\rightarrow$0,h=0$\rightarrow$1) of $s27$. For every line, we show its value in the form $a_1 \rightarrow a_2/a_3$, where a_1 is the value under the first pattern of the test, a_2 is the value under the second pattern if the circuit is fault free, and a_3 is the value under the second pattern if the circuit is faulty. We omit a_3 if it is equal to a_2. Both the transition fault g=1$\rightarrow$0 and the bridging fault (g=0,h=1) are detected by the test.

Given a parameter $n \geq 1$, we define for every transition fault g=$a \rightarrow a'$ up to n fault pairs (g=$a \rightarrow a'$,h_i=$a' \rightarrow a$) by selecting up to n lines h_i. We select the lines h_i out of a set of candidate lines H. The set H contains every line h_j such that there is no directed path from g to h_j or from h_j to g. This condition is imposed in order to avoid feedback bridging faults, which have additional detection requirements. We remove from H fanout branches and outputs of buffers. If $h_i \in H$ is a fanout branch of a stem h, or the output of a buffer with input h, then the fault (g=$a \rightarrow a'$,h_i=$a' \rightarrow a$) is equivalent to the fault (g=$a \rightarrow a'$,h=$a' \rightarrow a$), and there is no need to consider the first fault. We select n lines out of H randomly, and define fault pairs using every selected line.

For example, for $s27$ and the transition fault g=0$\rightarrow$1, the set of candidate lines $H = \{h_1,h_2,\cdots,h_{10}\}$ is shown in Figure 2. After removing fanout branches and outputs of buffers, we obtain $H = \{h_1,h_2,h_3,h_4,h_5,h_7,h_{10}\}$. For $n = 3$, we may select the fault pairs (g=0$\rightarrow$1, h_2=1$\rightarrow$0), (g=0$\rightarrow$1, h_5=1$\rightarrow$0), and (g=0$\rightarrow$1, h_7=1$\rightarrow$0).

By selecting n faults for every transition fault g=$a \rightarrow a'$, where $n \geq 1$, we define a set of target faults F_n. We define F_0 to be the set of all the transition faults in the circuit (in F_0 we do not pair bridging faults with the transition faults).

Other alternatives for the definition of fault pairs using bridging faults can use pairs including realistic bridging faults [17] or hard-to-detect bridging faults [18]. In addition, F_n can be defined dynamically during the test generation process.

3. Test generation

We perform test generation for F_n considering increasing values of n, $n = 0,1,\cdots$, until the target value of n is reached. We denote the test set generated for F_n by T_n.

We drive T_n by iteratively modifying tests in an initial test set T_0. Because of this the procedure does not guarantee the detection of all the detectable fault pairs in F_n. The experimental results presented in Section 4 demonstrate that the numbers of detections of transition faults grows proportionately to the numbers of tests. This demonstrates that n-detection test sets for transition faults can be generated while simultaneously targeting bridging faults without adversely affecting the size of the n-detection test sets.

The set of faults F_0 consists of transition faults, and the test set T_0 can be generated by any test generation procedure for transition faults. Since we decided to use broadside tests, the test generation procedure needs to be able to generate such tests. We obtain T_0 by fault simulating a large number of random broadside tests with fault dropping. We include in T_0 the tests that detect faults out of F_0 during the fault simulation process. Deterministic test generation can be used instead.

Before performing test generation for F_n, $n \geq 1$, we remove from F_n undetectable faults as follows. If a transition fault f_1 is not detected by T_0, we remove from F_n every fault pair that has f_1 as its first component. In addition, considering a fault pair (g=$a \rightarrow a'$,h_i=$a' \rightarrow a$), we check whether g stuck-at a can be propagated when $h_i = a$. If $h_i = a$ blocks all the paths through which g stuck-at a can be propagated to an output, we remove the fault from F_n.

For example, we consider the fault (g=1$\rightarrow$0,h_4=0$\rightarrow$1) in Figure 2. With $h_4 = 1$, the fault g stuck-at 1 cannot be propagated to an output. We therefore exclude this fault from F_n if it is included in it.

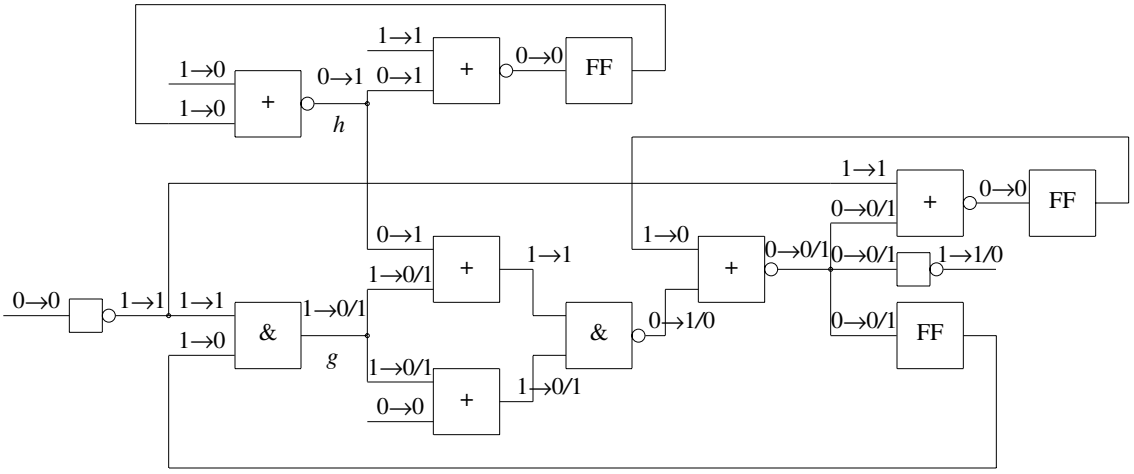

Figure 1: ISCAS-89 benchmark circuit s27

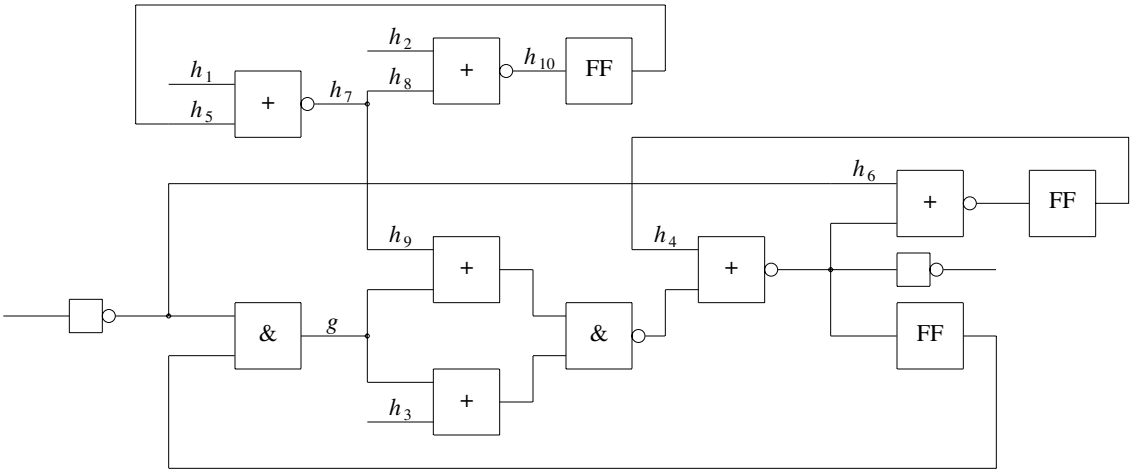

Figure 2: Example of candidate lines

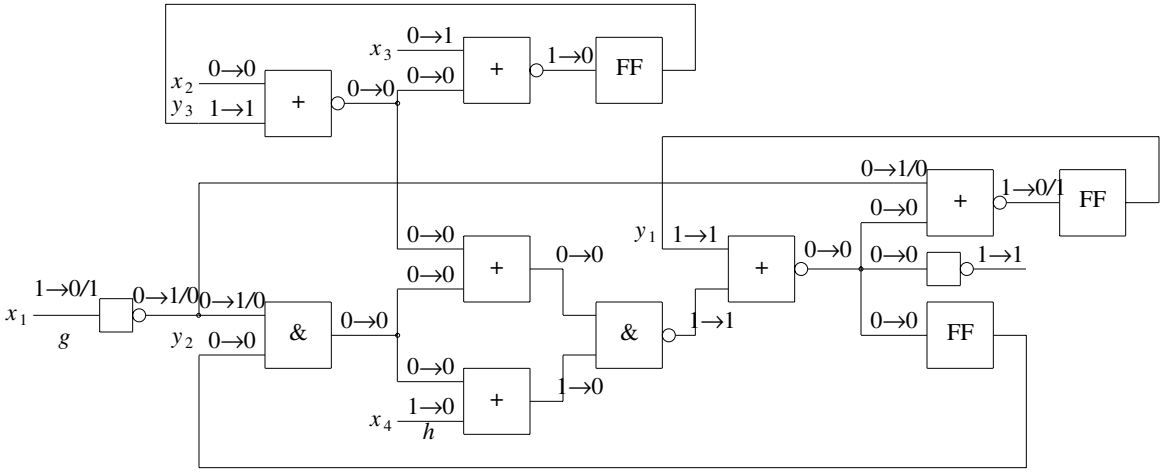

Figure 3: Example test

To generate a test set T_n for F_n, where $n \geq 1$, we first set $T_n = T_{n-1}$. We simulate the faults in F_n under T_n and remove detected faults from F_n. We then generate tests for the faults that remain in F_n as described next.

We use a simulation-based test modification procedure that iteratively modifies tests out of T_0 to detect faults in F_n. We consider every transition fault $g = a \rightarrow a'$ included in F_0 and detected by T_0. Let t_0 be the test in T_0 that detects $g = a \rightarrow a'$. For t_0, we consider every fault $(g = a \rightarrow a', h_i = a' \rightarrow a) \in F_n$, which is not already detected by T_n. We modify t_0 into a test t for $(g = a \rightarrow a', h_i = a' \rightarrow a)$ as follows.

The process of modifying t_0 generates several intermediate tests t before the final test is obtained. With every intermediate test t we associate the following parameters. We set $sadet = 1$ if the fault g stuck-at a is detected by the second pattern of t. We set $nsat$ equal to the number of requirements on g and h_i that are satisfied by t. The value of $nsat$ is incremented by one if $g = a$ under the first pattern of t, if $g = a'$ under the second pattern of t, if $h_i = a'$ under the first pattern of t, or if $h_i = a$ under the second pattern of t. For t_0 we have $sadet = 1$ and $nsat \geq 2$ (since t_0 satisfies the requirements for g). A test t that detects the target fault satisfies $sadet = 1$ and $nsat = 4$.

We first set $t = t_0$. We then complement the bits of t one at a time. After every bit i is complemented, we compute $sadet$ and $nsat$. If $sadet = 0$ or the value of $nsat$ decreased relative to its value before the complementation of bit i, we complement bit i again to undo the change.

We perform up to five iterations over all the bits of t, or until a test for the target fault is found. If a test t is found, it is added to T_n, all the faults in F_n are simulated under t, and detected faults are removed from F_n.

For illustration, we consider the fault $(g = 1 \rightarrow 0, h = 0 \rightarrow 1)$ in Figure 3. A broadside test for the transition fault $g = 1 \rightarrow 0$ is shown in Figure 3. The test does not detect the fault pair since h does not assume the value 0 under the first pattern, and it does not assume the value 1 under the second pattern. For the test shown in Figure 3, $sadet = 1$ and $nsat = 2$ (only g contributes to $nsat$).

We show the test of Figure 3 again in row t_0 of Table 1 using the symbols x_1, x_2, x_3, x_4 for the primary inputs, and y_1, y_2, y_3 for the present-state variables, as indicated in Figure 3. For t_0, $sadet = 1$ and $nsat = 2$. Modification of the test proceeds as shown in Table 1.

The first modification complements the value of y_1 under the first pattern. For the resulting test, $sadet = 1$ and $nsat = 2$. The second modification complements the value of y_2 under the first pattern. For the resulting test, $sadet = 1$ and $nsat = 2$. The third modification complements the value of y_3 under the first pattern. For the resulting test, $sadet = 1$ and $nsat = 2$. The fourth modification complements the value of x_1 under the first pattern. For the resulting test, $sadet = 1$ and $nsat = 1$.

Table 1: Example of test modification

	first pattern		second pattern		
	$y_1 y_2 y_3$	$x_1 x_2 x_3 x_4$	$x_1 x_2 x_3 x_4$	sadet	nsat
t_0	101	1001	0010	1	2
y_1	001	1001	0010	1	2
y_2	011	1001	0010	1	2
y_3	010	1001	0010	1	2
x_1	010	0001	0010	1	1
x_2	010	1101	0010	1	2
x_3	010	1111	0010	1	2
x_4	010	1110	0010	1	3
x_1	010	1110	1010	0	2
x_2	010	1110	0110	1	3
x_3	010	1110	0100	1	3
x_4	010	1110	0101	1	4

Since $nsat$ is reduced, x_1 is complemented again to obtain the previous test.

Complementation of x_2, x_3 and x_4 under the first pattern is accepted. After the complementation of x_4, $nsat$ increases to 3. Complementation of x_1 under the second pattern results in $sadet = 0$ and in the reduction of $nsat$. Therefore, x_1 is complemented again to restore the previous test. After complementing x_2, x_3 and x_4 under the second pattern, $sadet = 1$ and $nsat = 4$. The resulting test detects the target fault, and the process terminates.

4. Experimental results

We applied the test generation process described in Section 3 using $n = 0, 1, 2, \cdots, 10$. The results for $n \leq 4$ and $n = 10$ are shown in Tables 2 and 3. After the circuit name we show the value of n. Under column $flts$, for $n = 0$ we show the number of transition faults, and for $n \geq 1$ we show the number of faults in F_n after removing undetectable faults. Under column *init det* we show the number of faults out of F_n detected by T_{n-1}. For $n = 0$, this number is zero. Under column *tg det* we show the number of faults out of F_n that are detected after test generation. For $n = 0$, this is the number of faults detected by random broadside tests. Under column *tsts* we show the number of tests in T_n. Under column *rtio* we show the ratio $|T_n|/|T_0|$. Under column *ave trans* we show the average number of times a conventional transition fault is detected by the test set T_n. In computing the average we simulate a fault only until it is detected 10 times.

To provide an indication of the effectiveness of targeting fault pairs as proposed here, beyond the fact that increasing numbers of bridging faults are detected and that transition faults are detected increasing numbers of times, we perform robust simulation of path delay faults under the test sets T_n. We show the numbers of robustly detected path delay faults under column *pdf det*.

The following points can be seen from Tables 2 and 3. The number of fault pairs defined for $n = 1$ is typically smaller than the number of transition faults (reported for $n = 0$). This is due to the fact that some transition faults

Table 2: Results of test generation (I)

circuit	n	flts	init det	tg det	tsts	rtio	ave trans	pdf det
s208	0	416	0	321	58	1.00	3.26	19
s208	1	309	100	136	82	1.41	4.09	23
s208	2	613	273	297	101	1.74	4.50	24
s208	3	916	416	433	116	2.00	4.84	24
s208	4	1219	553	568	129	2.22	5.10	25
s208	10	3059	1409	1417	179	3.09	6.05	30
s298	0	596	0	487	80	1.00	4.47	100
s298	1	481	233	304	126	1.57	5.61	106
s298	2	954	597	625	143	1.79	5.88	113
s298	3	1434	900	926	161	2.01	6.19	113
s298	4	1914	1218	1230	170	2.12	6.28	113
s298	10	4777	3136	3143	221	2.76	6.89	115
s344	0	688	0	650	80	1.00	6.14	115
s344	1	640	351	430	128	1.60	7.73	137
s344	2	1280	845	888	156	1.95	8.16	140
s344	3	1919	1319	1353	184	2.30	8.41	144
s344	4	2559	1786	1808	199	2.49	8.54	145
s344	10	6419	4608	4620	279	3.49	8.97	155
s382	0	764	0	599	77	1.00	4.19	104
s382	1	590	273	398	144	1.87	6.00	127
s382	2	1183	782	847	187	2.43	6.52	130
s382	3	1773	1205	1252	219	2.84	6.81	132
s382	4	2362	1643	1668	240	3.12	6.91	133
s382	10	5903	4194	4217	330	4.29	7.37	135
s386	0	772	0	612	112	1.00	3.98	116
s386	1	595	214	285	160	1.43	4.97	118
s386	2	1191	505	534	184	1.64	5.29	122
s386	3	1790	782	801	200	1.79	5.48	122
s386	4	2385	1055	1108	227	2.03	5.77	128
s386	10	5969	2721	2736	289	2.58	6.27	129
s400	0	800	0	617	77	1.00	4.06	99
s400	1	605	311	432	138	1.79	5.63	114
s400	2	1216	778	858	190	2.47	6.33	118
s400	3	1825	1225	1279	223	2.90	6.63	125
s400	4	2434	1672	1704	246	3.19	6.79	128
s400	10	6092	4338	4351	336	4.36	7.26	136
s420	0	840	0	607	130	1.00	3.52	26
s420	1	580	178	249	172	1.32	4.28	27
s420	2	1168	486	541	211	1.62	4.88	29
s420	3	1746	783	816	238	1.83	5.15	32
s420	4	2328	1056	1098	272	2.09	5.40	33
s420	10	5845	2795	2820	387	2.98	6.10	38
s510	0	1020	0	917	131	1.00	4.33	133
s510	1	897	288	353	189	1.44	5.39	137
s510	2	1798	646	690	228	1.74	5.98	139
s510	3	2701	1026	1065	259	1.98	6.39	141
s510	4	3605	1410	1442	284	2.17	6.68	141
s510	10	9012	3538	3555	381	2.91	7.47	144
s526	0	1052	0	680	123	1.00	3.54	130
s526	1	663	271	379	196	1.59	4.45	132
s526	2	1333	700	764	241	1.96	4.86	133
s526	3	2001	1098	1147	276	2.24	5.09	135
s526	4	2672	1509	1554	306	2.49	5.27	135
s526	10	6692	3956	3984	455	3.70	5.75	135
s641	0	1280	0	1213	171	1.00	7.13	263
s641	1	1199	749	986	321	1.88	8.34	342
s641	2	2404	1881	2018	421	2.46	8.72	355
s641	3	3610	2960	3063	492	2.88	8.91	381
s641	4	4814	4030	4097	538	3.15	9.01	396
s641	10	12030	10361	10389	764	4.47	9.27	412

Table 3: Results of test generation (II)

circuit	n	flts	init det	tg det	tsts	rtio	ave trans	pdf det
s820	0	1640	0	1318	268	1.00	3.69	258
s820	1	1299	410	570	397	1.48	4.82	280
s820	2	2597	1050	1150	476	1.78	5.27	288
s820	3	3887	1629	1706	541	2.02	5.59	292
s820	4	5182	2216	2294	598	2.23	5.85	294
s820	10	12940	5753	5821	872	3.25	6.78	304
s953	0	1906	0	1804	256	1.00	5.49	548
s953	1	1797	638	836	392	1.53	6.89	630
s953	2	3577	1601	1744	492	1.92	7.61	663
s953	3	5363	2489	2624	577	2.25	8.03	682
s953	4	7149	3402	3479	636	2.48	8.24	697
s953	10	17823	8860	8929	884	3.45	8.89	748
s1196	0	2392	0	2366	459	1.00	6.31	660
s1196	1	2348	1243	1647	755	1.64	7.74	854
s1196	2	4694	3042	3348	985	2.15	8.39	928
s1196	3	7044	4871	5082	1149	2.50	8.68	1006
s1196	4	9401	6592	6789	1305	2.84	8.89	1046
s1196	10	23508	17032	17169	1870	4.07	9.39	1187
s1423	0	2846	0	2494	256	1.00	6.62	543
s1423	1	2483	1463	1972	514	2.01	7.84	714
s1423	2	4971	3742	4014	666	2.60	8.13	787
s1423	3	7458	5873	6075	794	3.10	8.31	807
s1423	4	9943	8005	8167	890	3.48	8.40	819
s1423	10	24859	20754	20856	1330	5.20	8.63	890
s1488	0	2976	0	2727	327	1.00	5.72	342
s1488	1	2710	1190	1322	419	1.28	6.44	358
s1488	2	5416	2550	2633	480	1.47	6.79	365
s1488	3	8116	3880	3946	531	1.62	7.08	370
s1488	4	10824	5257	5334	587	1.80	7.33	380
s1488	10	27058	13463	13499	773	2.36	7.88	399
s5378	0	10590	0	9588	714	1.00	7.43	2201
s5378	1	9577	7145	8302	1364	1.91	8.34	2782
s5378	2	19156	16125	16736	1768	2.48	8.57	3023
s5378	3	28738	24742	25252	2137	2.99	8.71	3221
s5378	4	38315	33328	33753	2418	3.39	8.79	3317
s5378	10	95774	85127	85388	3714	5.20	8.98	3686
s9234	0	18468	0	13253	978	1.00	4.90	1835
s9234	1	13242	7085	9539	2263	2.31	6.55	2384
s9234	2	26479	18223	19473	3038	3.11	6.89	2497
s9234	3	39714	28557	29510	3662	3.74	7.05	2566
s9234	4	52951	38740	39530	4191	4.29	7.13	2595
s9234	10	132401	99837	100374	6461	6.61	7.30	2778
s13207	0	26358	0	20504	1126	1.00	6.17	2921
s13207	1	20494	12316	15258	2629	2.33	7.28	3149
s13207	2	40988	29427	30937	3653	3.24	7.52	3183
s13207	3	61484	45463	46707	4515	4.01	7.63	3206
s13207	4	81975	61599	62722	5259	4.67	7.70	3231
s13207	10	204944	158534	159406	8985	7.98	7.85	3282

are not detected by T_0, and we do not define fault pairs based on undetected transition faults. In addition, undetectable fault pairs are removed from F_1.

The number of detected fault pairs increases significantly due to test generation for $n = 1$ and $n = 2$. For example, we consider $s208$. Before test generation for $n = 1$, the initial test set T_1 is equal to T_0. This test set detects 100 fault pairs, or 100 bridging faults. After test generation for $n = 1$, the test set T_1 detects 136 bridging faults. Considering $n = 2$, the initial test set T_2 (equal to T_1) detects 273 fault pairs. After test generation, T_2 detects 297 fault pairs. The increase in the number of detected faults obtained for $n = 1$ and $n = 2$ is sometimes even higher for other circuits.

For $n > 2$, the increase in the number of detected bridging faults due to test generation is lower. For example, for $s208$, test generation for F_3 increases the number of detected bridging faults from 416 faults detected by the initial test set T_3, to 433 faults detected by the test set T_3 after test generation. For $n = 10$, the number of detected faults increases from 1409 to 1417.

The increase in test set size and in the number of detected bridging faults results in an increase in the average number of detections of transition faults. The number of detected path delay faults increases as well. The test set size grows moderately with n. This is unlike n-detection test generation, where the increase in test set size is approximately linear with n. This demonstrates that the proposed method to target bridging faults while generating n-detection test sets for transition faults

achieves n-detections of transition faults without affecting the size of the test set adversely.

In Table 4 we show the increase in test generation time as n is increased. We denote the test generation time for a given value of n by RT_n. The run time is cumulative, i.e., RT_n includes the test generation time for $F_1, F_2, \cdots, F_n$. We report in Table 4 the value of RT_n/RT_1, for $1 \leq n \leq 10$, for several circuits.

Table 4: Increase in test generation time with n

n	s641	s1196	s1423	s1488	s5378
1	1.00	1.00	1.00	1.00	1.00
2	1.65	1.95	1.69	1.50	2.17
3	2.54	3.28	2.65	2.24	3.82
4	3.69	5.03	3.86	3.21	5.95
5	5.07	7.18	5.31	4.42	8.55
6	6.66	9.75	7.01	5.87	11.62
7	8.50	12.74	8.96	7.57	15.12
8	10.58	16.10	11.15	9.50	19.06
9	12.92	19.92	13.61	11.67	23.44
10	15.48	24.13	16.29	14.08	28.23

Beyond a certain value of n, every transition fault $g = a \rightarrow a'$ is paired with every bridging fault that shares with it the requirement to detect the fault g stuck-at a, and no additional fault pairs are defined. We increased n in multiples of 2 for several circuits in order to reach this point. The importance of this point is that the test set detects all the bridging faults that can be detected together with transition faults by broadside tests. We perform test generation only for the final value of n (and not for every value of n between 1 and the maximum value). For $s1423$, $n = 1024$ resulted in 1563546 fault pairs, of which 914915 are detected by T_0. Test generation resulted in 7026 tests that detect 1380408 fault pairs. The average number of detections of transition faults was 8.79.

5. Concluding remarks

We investigated the generation of n-detection test sets for transition faults by pairing each transition fault with n four-way bridging faults. A test for a fault pair must detect both faults when they are present in the circuit individually. To facilitate test generation, we selected the fault pairs such that both the transition fault and the bridging fault in a pair require the detection of the same stuck-at fault. The stuck-at fault (as well the transition and the bridging faults) was detected by the second pattern of a broadside test. The advantage of this approach is that it ensures the detection of bridging faults that would not be targeted, and may not be detected, during n-detection test generation for transition faults. Experimental results demonstrated that the test set size grows moderately with n. In addition, we showed that increasing n increases the coverage of path delay faults, which were not targeted.

References

[1] S. C. Ma, P. Franco and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results", Intl. Test Conf., 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] I. Pomeranz and S. M. Reddy, "Stuck-At Tuple-Detection: A Fault Model based on Stuck-At Faults for Improved Defect Coverage", 16th VLSI Test Symp., April 1998, pp. 289-294.

[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", Intl. Test Conf., 1998, pp. 184-193.

[5] I. Pomeranz and S. M. Reddy, "On n-Detection Test Sets and Variable n-Detection Test Sets for Transition Faults", 17th VLSI Test Symp., April 1999, pp. 173-179.

[6] M. R. Grimaila, S. Lee, J. Dworak, K. M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, L.-C. Wang and M. R. Mercer, "REDO - Random Excitation and Deterministic Observation - First Commercial Experiment", VLSI Test Symp., 1999, pp. 268-274.

[7] J. Dworak, M. R. Grimaila, S. Lee, L.-C. Wang and M. R. Mercer, "Enhanced DO-RE-ME based Defect Level Prediction using Defect Site Aggregation-MPG-D", Intl. Test Conf., Oct. 2000, pp. 930-939.

[8] I. Pomeranz and S. M. Reddy, "Definitions of the Numbers of Detections of Target Faults and their Effectiveness in Guiding Test Generation for High Defect Coverage", Intl. Design Autom. and Test in Europe Conf., March 2001, pp. 504-508.

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

[10] 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", Intl. Test Conf., Sept. 2003, pp. 1031-1040.

[11] 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", 22nd VLSI Test Symp., April 2004, pp. 23-28.

[12] Z. Barzilai and B. Rosen, "Comparison of AC Self-Testing Procedures", 1983 Intl. Test Conf., pp. 89-94.

[13] S. Sengupta et. al., "Defect-Based Tests: A Key Enabler for Successful Migration to Structural Test", Intel Technology Journal, Q.1, 1999.

[14] V. Krishnaswamy, A. B. Ma, P. Vishakantaiah, "A Study of Bridging Defect Probabilities on a Pentium (TM) 4 CPU", Intl. Test Conf., 2001, pp. 688-695.

[15] J. Savir and S. Patil, "Broad-Side Delay Test", IEEE Trans. on Computer-Aided Design, Aug. 1994, pp. 1057-1064.

[16] J. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, "Transition Fault Simulation", IEEE Design & Test, April 1987, pp. 32-38.

[17] F. J. Ferguson and J. P. Shen, "A CMOS Fault Extractor for Inductive Fault Analysis", IEEE Trans. on Computer-Aided Design, Nov. 1988, pp. 1181-1194.

[18] I. Pomeranz, S. M. Reddy and S. Kundu, "On the Characterization of Hard-to-Detect Bridging Faults", Intl. Design Autom. and Test in Europe Conf., March 2003, pp. 1012-1017.