Switching Activity of Faulty Circuits in Presence of Multiple Transition Faults

Irith Pomeranz, *Fellow, IEEE*

***Abstract*—Excessive switching activity in the fault-free circuit can cause a fault-free circuit to fail because of increased delays. For the same reason, excessive switching activity in a faulty circuit can cause a delay fault to escape detection. This paper observes that the switching activity is higher in the presence of multiple delay faults with higher multiplicities (larger numbers of single faults that are present in the circuit together). The challenge in addressing multiple faults is related to their large number. This paper addresses this challenge by an iterative procedure that is applied to transition faults under a low-power broadside test set, and has two subprocedures: 1) the first subprocedure finds multiple transition faults with excessive faulty switching activity and 2) the second subprocedure modifies the test set so as to avoid excessive faulty switching activity for the faults found by the first subprocedure. With every additional iteration there are fewer multiple transition faults that exhibit excessive faulty switching activity. The experimental results for benchmark circuits demonstrate the levels of excessive faulty switching activity in the presence of multiple transition faults, and the possibility of reducing or even eliminating it after a small number of iterations.**

***Index Terms*—Fault-free switching activity, faulty switching activity, full-scan circuit, low-power test set, multiple transition faults.**

I. Introduction

LOW-POWER test generation procedures for scan circuits prevent excessive switching activity, and power dissipation, during the application of scan-based tests [1]–[24]. A low-power test generation procedure may address the scan shift cycles or functional capture cycles of scan-based tests. The importance of the switching activity during functional capture cycles results from the fact that faults are activated and propagated during these clock cycles. Specifically, for a broadside (launch-on-capture) test, there are two functional capture cycles between a scan-in and scan-out operation. Fault effects that reach the primary outputs or next-state variables during the second functional capture cycle are observed. Accordingly, for the discussion in this paper, the switching activity of a broadside test is defined as the percentage of lines that make a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition during the second functional capture cycle. The two functional capture cycles of a broadside test are illustrated by Fig. 1. Fig. 1 shows a fanout stem with two branches inside the combinational logic of the circuit. Each one of the three lines shown in Fig. 1 makes a $0 \rightarrow 1$ transition during the second functional capture cycle of the test, and thus contributes to its switching activity. As in most low-power test generation procedures, the switching activity is used in this paper for measuring the power dissipation of a test. This is based on the expectation that the switching activity is the most dominant contributor to the power dissipation. More accurate metrics that take hazards into consideration, or incorporate timing information, can be used instead of the switching activity. In addition, the circuit can be partitioned into subcircuits based on the power grid, and the switching activity (or a different metric of power dissipation) can be computed for each subcircuit individually. The procedure described in this paper can be applied with any metric of the power dissipation.

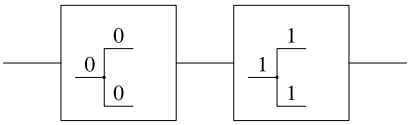

Fig. 1. Broadside test.

Excessive switching activity during the second functional capture cycle of a broadside test can result in voltage drops that increase the delays in the circuit. As a result, the circuit may slow down. Consequently, a fault-free circuit that would operate correctly during functional operation may appear to be faulty during test application, resulting in unnecessary yield loss [1], [2].

The switching activity discussed thus far occurs in the fault-free circuit, and referred to as the fault-free switching activity. Excessive switching activity can also occur in a faulty circuit. The faulty switching activity for a fault f under a broadside test t is defined as the percentage of lines that make a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition during the second functional capture cycle of t in the presence of f.

The faulty switching activity was considered in [25] and [26]. The fact that excessive switching activity can cause delays in the circuit to increase, and thus cause a fault-free circuit to appear to be faulty, also implies that a faulty circuit can escape detection by a test with excessive faulty switching activity. This occurs, e.g., when the propagation of a fault effect from the site of a delay

Manuscript received October 28, 2018; revised January 2, 2019; accepted February 24, 2019. Date of publication February 27, 2019; date of current version March 18, 2020. This work was supported in part by NSF under Grant CCF-1714147. This paper was recommended by Associate Editor N. Nicolici.

The author is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: pomeranz@ecn.purdue.edu).

Digital Object Identifier 10.1109/TCAD.2019.2902326

fault f requires a transition on a particular line g. Excessive switching activity in the faulty circuit can delay the transition on line g, preventing the fault f from being detected.

The discussion in [25] and [26] considered single transition faults under low-power broadside tests. For this discussion, an upper bound on the allowed switching activity is denoted by swa_{max}. Both the fault-free and faulty switching activity is required not to exceed swa_{max} in order to avoid unnecessary yield loss as well as test escapes. The low-power test set considered in [25] and [26] satisfies this bound for the fault-free circuit. The discussion in [25] and [26] yielded the following results.

1) Values of the faulty switching activity that are higher than swa_{max} were obtained for most of the benchmark circuits considered.
2) Significant numbers of tests were shown to suffer from excessive faulty switching activity.
3) Significant numbers of faults were shown to cause excessive faulty switching activity.
4) When tests with excessive faulty switching activity were removed from the test set, the fault coverage loss was significant.
5) To avoid the fault coverage loss, a postprocessing procedure was developed in [26] to modify a low-power test set so as to avoid excessive faulty switching activity. The procedure modifies every test that has excessive faulty switching activity. For such a test, it creates a safety margin for the faulty switching activity by reducing the fault-free switching activity. After several iterations of modification with an increasing safety margin, the test can accommodate an increased faulty switching activity without exceeding the bound given by swa_{max}.

This paper observes that increased faulty switching activity is more likely to be obtained for a multiple transition fault, where several single transition faults are present in the faulty circuit together, than for a single transition fault. Moreover, even if a sufficient safety margin is created for a low-power test set based on single transition faults, the safety margin may not be sufficient in the presence of multiple transition faults. Multiple faults are prevalent in early stages of the yield improvement process for a new technology, and they are likely to occur in the state-of-the-art technologies with a high device density. The challenge in considering multiple transition faults is that their number is too large to consider all of them. In addition, without simulation, it is not possible to predict accurately which multiple transition faults will suffer from excessive faulty switching activity. This paper addresses these challenges by an iterative procedure that has two subprocedures, which are themselves iterative.

1) The first subprocedure finds multiple transition faults with excessive faulty switching activity. Such a subprocedure was not needed in [25] and [26], where only single transition faults are considered.
2) The second subprocedure modifies a given test set to avoid excessive faulty switching activity for the multiple transition faults found by the first subprocedure. This is accomplished by extending the procedure from [26] to consider multiple transition faults.

The first subprocedure does not find all the multiple transition faults of interest in one iteration. Successive iterations, and the modification of the test set, increase the confidence that the test set will avoid excessive faulty switching activity in general, even for faults that were not considered yet. This is supported by experimental results showing that it becomes more difficult to find multiple transition faults with excessive faulty switching activity as more iterations are performed.

This paper does not include silicon experiments to verify that excessive faulty switching activity can cause test escapes. Instead, it relies on the following arguments.

1) Silicon experiments in [1] and [2] demonstrate the possibility of fault-free circuits failing because of excessive fault-free switching activity. The same mechanisms of increased delays that explain the failure of fault-free circuits also support the possibility of faulty circuits escaping detection.
2) The space of possible test sets offers a vast choice for a suitable test set. Avoiding tests with excessive faulty switching activity improves the confidence that the test set will perform its task without giving up any quality metrics, such as the fault coverage, the fault-free switching activity, or the number of tests.

This paper is organized as follows. Details from [25] and [26] are reviewed in Section II. An example of a multiple transition fault that can potentially escape detection because of excessive faulty switching activity is given in Section III. The procedure for addressing excessive faulty switching activity in the presence of multiple transition faults is discussed in Section IV. The experimental results are presented in Section V.

II. Faulty Switching Activity

This section reviews details of the procedures from [25] and [26] that are important for the discussion in this paper.

An example illustrating how the presence of a single transition fault can increase the faulty switching activity is given in [25]. An example involving a double fault is given in Section III. This example is more relevant to the discussion in this paper.

The low-power test set considered in [25] and [26] is generated by the procedure described in [20]. The same test set is considered in this paper. The low-power test generation procedure from [20] is guided by a test set that consists of functional broadside tests. A functional broadside test is a two-cycle snapshot of state-transitions that can occur during functional operation. Therefore, the switching activity of a functional broadside test is guaranteed to be possible during functional operation. The maximum switching activity of a functional broadside test is used as an upper bound on the switching activity allowed for a low-power test in [20]. The same bound is used in [25] and [26], as well as this paper, for the bound swa_{max}. Thus, swa_{max} does not exceed the switching activity that is possible during functional operation, and the same applies to every low-power test.

The following example from [26] illustrates how tests in a low-power test set are modified to reduce their faulty switching

TABLE I
EXAMPLE OF MODIFICATION PROCEDURE FROM [26]

	iter0		iter1			iter2		
i	$swa(t_i)$	$swa(t_i, f_i)$	τ_i	$swa(t_i)$	$swa(t_i, f_i)$	τ_i	$swa(t_i)$	$swa(t_i, f_i)$
356	2394	2495	2345	2288	-			
359	2417	2452	2411	2370	-			
360	2420	2449	2417	2401	-			
361	2420	2451	2415	2380	-			
362	2421	2473	2394	2389	-			
363	2424	2481	2389	2372	-			
364	2425	2475	2396	2390	-			
365	2428	2464	2410	2400	2471	2375	2373	-
366	2429	2459	2416	2367	-			
367	2446	2480	2412	2383	-			

activity. The circuit under consideration is the benchmark circuit *s*5378. The test set is the one from [20]. The test set contains 368 tests, and the bound on the switching activity is $\text{swa}_{\text{max}} = 2446$. The target faults are single transition faults. Column *i* of Table I shows the indices of ten tests with excessive faulty switching activity. Column iter0 of Table I shows the fault-free switching activity $\text{swa}(t_i)$ for every one of these tests, and its maximum faulty switching activity $\text{swa}(t_i, f_i)$, which is obtained for a single transition fault denoted by f_i.

If the ten tests with excessive faulty switching activity are removed from the test set, the fault coverage decreases from 92.01% to 91.86%. Instead, the procedure from [26] modifies the tests so as to create a safety margin based on the fault-free switching activity. The safety margin that the procedure from [26] uses for a test t_i is equal to $\text{swa}(t_i, f_i) - \text{swa}(t_i)$. Accordingly, its target for the fault-free switching activity of t_i is equal to $\tau_i = \text{swa}_{\text{max}} - (\text{swa}(t_i, f_i) - \text{swa}(t_i))$.

The procedure reduces the fault-free switching activity of a test t_i by attempting to complement its bits one by one in a random order. It accepts to complement a bit if the fault coverage of the test set is not reduced, and the fault-free switching activity of the test is not increased. The target of τ_i for the fault-free switching activity is typically achieved after complementing a small number of bits. Otherwise, the procedure considers all the bits a constant number of times.

The safety margin given by τ_i is conservative, and it is not always sufficient for ensuring that the faulty switching activity does not exceed swa_{max}. Therefore, the modification procedure is applied iteratively as long as it reduces the faulty switching activity of at least one test with excessive faulty switching activity.

Column iter1 of Table I shows the results of the first iteration of the modification procedure for *s*5378. For nine out of ten tests, the safety margin is sufficient for ensuring that the faulty switching activity does not exceed swa_{max} for any fault. This is indicated in Table I by the dashes under column iter1 subcolumn $\text{swa}(t_i, f_i)$. For the remaining test, an additional iteration is needed. This iteration prevents excessive faulty switching activity for the remaining test as shown under column iter2 of Table I.

In the case of *s*5378, removing the tests with excessive faulty switching activity from the initial low-power test set reduces the fault coverage by 0.15%. For other circuits the effect on the fault coverage is more significant. For example, for benchmark circuit *s*9234, removing tests with excessive faulty switching activity reduces the fault coverage of the initial test set by 12.68%. For benchmark circuit *wb_dma* the fault coverage is reduced by 13.85%. The effect is expected to be even more significant when multiple transition faults are considered as suggested in this paper. Therefore, it is important to modify the tests as suggested in [26], and maintain the fault coverage.

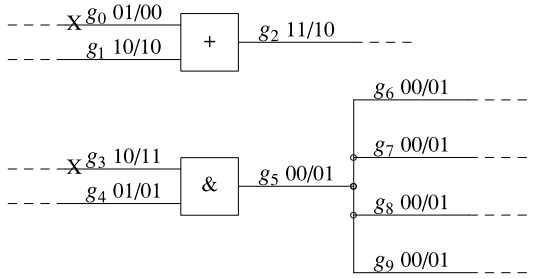

Fig. 2. Double transition fault.

III. MULTIPLE TRANSITION FAULTS

An example of a test for a double transition fault is shown in Fig. 2. Line values are shown in Fig. 2 using the following format. A pair of values corresponds to the two functional capture cycles of the test in the fault-free or faulty circuit. Pairs of fault-free/faulty values are shown in this order.

The double fault in Fig. 2 delays the rising transition on line g_0, and the falling transition on line g_3. These lines are marked with an *X* in Fig. 2. The fault effects can be seen from the pair of values 01 on g_0 in the fault-free circuit, corresponding to a rising transition, and the pair of values 00 in the faulty circuit, indicating that the transition is delayed. Similarly, line g_3 makes a falling transition in the fault-free circuit, but not in the faulty circuit.

The falling transition on line g_1 propagates the fault effect from line g_0. Similarly, the rising transition on line g_4 propagates the fault effect from line g_3. New falling and rising transitions appear in the faulty circuit on lines g_2, g_5, g_6, g_7, g_8, and g_9.

Overall, four lines in the fault-free circuit carry transitions and contribute to the fault-free switching activity. In the faulty circuit, there are eight lines with transitions that contribute to the faulty switching activity. This example illustrates that the faulty switching activity may be higher than the fault-free switching activity.

Suppose that the double transition fault in Fig. 2 is detected by propagating the fault effect from line g_2 to an output.

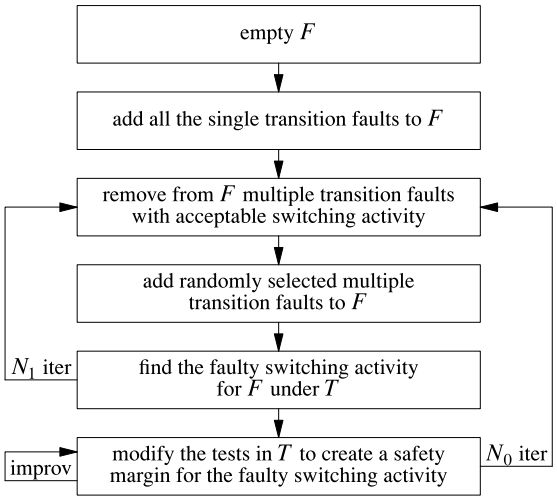

Fig. 3. Procedure for multiple transition faults.

Suppose in addition that the increased switching activity in the faulty circuit causes delays in the faulty circuit to increase. If the transition on line g_1 is delayed, the line will carry the pair of values 11 in the faulty circuit. With this pair of values, the fault effect from line g_0 is not propagated to line g_2. As a result, the fault escapes detection.

The difference between the fault-free and faulty switching activity can potentially be higher with faults of higher multiplicities. In Fig. 2, a third transition fault on a line g_{10} that drives a gate with fanout without being detected, similar to g_3, will increase the faulty switching activity further. It is possible to search for single transition faults that are propagated by transitions, and create new transitions, and use them to define multiple transition faults. This approach suffers from several limitations.

1) Not all the multiple transition faults with excessive faulty switching activity may be found in this way.
2) After the test set is modified to create larger safety margins for the faulty switching activity, this approach cannot be repeated to find new multiple transition faults with excessive faulty switching activity.

The procedure described in the next section avoids these limitations.

IV. Addressing Excessive Faulty Switching Activity of Multiple Transition Faults

An iterative procedure for addressing excessive faulty switching activity of multiple transition faults is discussed in this section. The procedure is illustrated by Fig. 3.

A. Overview

The procedure is applied to a low-power broadside test set T for single transition faults. The fact that the test set is low-power implies that the fault-free switching activity of the tests in T does not exceed the bound swa_{max}.

The set of target faults is denoted by F. The set of single transition faults is added to F once, at the beginning, and kept in it. This ensures that, throughout all the iterations, the test set T continues to detect all the detectable single transition faults, and their faulty switching activity does not exceed the bound given by swa_{max}.

The procedure from Fig. 3 has two subprocedures, which are also applied iteratively. The numbers of iterations, or the termination condition, are shown in the figure and discussed later. In the first iterative subprocedure, the procedure from Fig. 3 selects multiple transition faults that will be added to F. The goal is to include in F as many multiple transition faults as possible with excessive faulty switching activity, without considering inordinate numbers of faults.

In the second iterative subprocedure, tests with excessive faulty switching activity in the test set T are modified so as to reduce the faulty switching activity to the acceptable level of swa_{max}. For this purpose, the procedure described in [26] is extended to consider multiple transition faults. The target faults are the ones included in F by the first subprocedure. The modification is performed iteratively, with increasing safety margins, until no further improvements are obtained. In this case, the variable improv in Fig. 3 becomes zero. The subprocedure typically terminates when none of the faults in F creates excessive faulty switching activity under T.

The selection of multiple transition faults, and the modification of the test set, are repeated to identify new multiple transition faults with excessive faulty switching activity, and modify the test set further to avoid excessive faulty switching activity. The iterations compensate for the fact that it is not possible to identify all the multiple transition faults with excessive faulty switching activity in one iteration. The modification of the test set ensures that the fault-free switching activity provides increased safety margins to avoid excessive faulty switching activity. As a result, it becomes progressively more difficult to find multiple transition faults with excessive faulty switching activity. Thus, the test set becomes more and more immune to excessive faulty switching activity.

The number of iterations of the two subprocedures is given by a parameter that is denoted by N_0. The details of the subprocedure that selects multiple transition faults are described next. This is followed by a discussion of the parameters N_0 and N_1 in Fig. 3.

B. Selecting Multiple Transition Faults

Before the first iteration of the procedure from Fig. 3, only single transition faults are included in F. The step in Fig. 3 where multiple transition faults are removed from F does not change the set F in the first iteration. This step is discussed later. In the next step, the first subprocedure adds multiple transition faults to F, as follows.

For a circuit with L lines, the number of single transition faults is $2L$. The subprocedure adds to F the same number of faults of multiplicity m, for $2 \leq m \leq M$, where M is a constant. As a result, the size of F is increased by $2L(M-1)$.

To form a transition fault of multiplicity m, the first subprocedure selects m single transition faults randomly, and considers them as present in the circuit together. The only constraint that the procedure imposes is not to select two single transition faults that are associated with the same line.

TABLE II
EXPERIMENTAL RESULTS ($swa_{max} < 26\%$)

							ff swa			fy swa									
circuit	sv	it0	sub	it1	tests	f.c.(%)	max(%)	ave(%)	tests	swa1(%)	swa2(%)	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	ntime
s15850	597	0	0	0	590	90.41	13.50	12.65	322	15.08	15.18	2204	4226	5934	7528	8926	10424	11426	12466	13.35
s15850	597	1	1	3	590	90.51	13.26	12.20	0	-	-	-	-	-	-	-	-	-	-	235.44
s15850	597	2	0	1	590	90.55	13.26	12.20	41	-	14.37	-	-	3	2	5	10	8	21	248.77
s15850	597	2	1	2	590	90.55	13.26	12.18	0	-	-	-	-	-	-	-	-	-	-	274.58
s15850	597	3	0	2	590	90.44	13.26	12.18	69	-	14.29	-	2	2	3	9	16	8	28	301.15
s15850	597	3	1	2	590	90.44	13.24	12.14	0	-	-	-	-	-	-	-	-	-	-	342.95
s15850	597	4	0	3	590	90.42	13.24	12.14	48	-	14.07	-	-	-	6	5	7	10	17	382.52
s15850	597	4	1	2	590	90.42	13.24	12.12	0	-	-	-	-	-	-	-	-	-	-	411.00
s13207	669	0	0	0	186	85.36	18.16	17.72	98	20.57	20.57	1422	2710	3945	4939	5719	6583	7450	8090	6.61
s13207	669	1	1	2	186	85.63	18.01	17.44	0	-	-	-	-	-	-	-	-	-	-	69.98
s13207	669	2	0	1	186	85.69	18.01	17.44	19	-	18.46	-	-	-	1	4	4	6	5	76.30
s13207	669	2	1	1	186	85.70	18.01	17.43	0	-	-	-	-	-	-	-	-	-	-	88.46
s13207	669	3	0	2	186	85.64	18.01	17.43	20	-	18.67	-	1	1	1	-	4	5	11	100.66
s13207	669	3	1	2	186	85.64	18.01	17.40	0	-	-	-	-	-	-	-	-	-	-	116.47
s13207	669	4	0	3	186	85.49	18.01	17.40	15	-	18.59	-	-	-	3	4	4	4	6	139.75
s13207	669	4	1	1	186	85.49	17.99	17.39	0	-	-	-	-	-	-	-	-	-	-	152.92
wb_dma	523	0	0	0	746	99.22	22.26	20.41	461	26.48	29.66	824	1576	2236	2835	3260	3787	4109	4474	38.80
wb_dma	523	1	1	4	746	99.23	22.19	18.90	0	-	-	-	-	-	-	-	-	-	-	171.38
wb_dma	523	2	0	1	746	99.23	22.19	18.90	83	-	25.34	-	1	-	6	13	9	15	14	209.95
wb_dma	523	2	1	3	746	99.23	22.19	18.82	0	-	-	-	-	-	-	-	-	-	-	233.88
wb_dma	523	3	0	2	746	99.23	22.19	18.82	81	-	25.04	-	1	1	7	8	9	13	19	311.00
wb_dma	523	3	1	4	746	99.23	22.19	18.75	0	-	-	-	-	-	-	-	-	-	-	336.03
wb_dma	523	4	0	3	746	99.23	22.19	18.75	91	-	25.20	-	-	2	4	9	14	20	27	451.71
wb_dma	523	4	1	3	746	99.23	22.19	18.64	0	-	-	-	-	-	-	-	-	-	-	477.13
tv80	359	0	0	0	1161	96.88	24.90	15.00	33	28.39	28.61	594	1125	1707	2074	2613	3056	3327	3780	14.27
tv80	359	1	1	3	1161	96.88	24.24	14.96	0	-	-	-	-	-	-	-	-	-	-	28.21
tv80	359	2	0	1	1161	96.86	24.24	14.96	13	-	27.89	-	-	-	1	4	5	1	3	42.62
tv80	359	2	1	2	1161	96.86	24.24	14.95	0	-	-	-	-	-	-	-	-	-	-	48.05
tv80	359	3	0	2	1161	96.90	24.24	14.95	14	-	25.44	-	-	-	1	2	2	4	5	76.61
tv80	359	3	1	2	1161	96.90	23.89	14.95	0	-	-	-	-	-	-	-	-	-	-	82.02
tv80	359	4	0	3	1161	96.88	23.89	14.95	11	-	25.43	-	-	-	3	1	2	1	5	125.33
tv80	359	4	1	3	1161	96.88	23.74	14.94	0	-	-	-	-	-	-	-	-	-	-	131.43
simple_spi	131	0	0	0	220	98.99	25.18	20.42	88	28.69	30.37	221	423	590	717	873	975	1114	1180	16.87
simple_spi	131	1	1	3	220	98.99	24.19	19.14	0	-	-	-	-	-	-	-	-	-	-	34.96
simple_spi	131	2	0	1	220	99.01	24.19	19.14	12	-	27.33	-	1	-	1	1	2	1	2	52.12
simple_spi	131	2	1	2	220	99.01	24.19	19.02	0	-	-	-	-	-	-	-	-	-	-	56.74
simple_spi	131	3	0	2	220	98.98	24.19	19.02	15	-	27.02	-	-	-	3	3	3	2	12	90.46
simple_spi	131	3	1	1	220	99.01	24.19	18.92	0	-	-	-	-	-	-	-	-	-	-	94.08
simple_spi	131	4	0	3	220	99.01	24.19	18.92	18	-	26.75	-	-	-	1	2	7	6	5	144.22
simple_spi	131	4	1	2	220	99.01	24.19	18.75	0	-	-	-	-	-	-	-	-	-	-	148.60
b15	447	0	0	0	1400	97.82	25.43	15.49	31	28.66	28.70	501	998	1418	1837	2164	2660	2995	3326	22.45
b15	447	1	1	3	1400	97.82	24.61	15.46	0	-	-	-	-	-	-	-	-	-	-	34.15
b15	447	2	0	1	1400	97.81	24.61	15.46	6	-	25.78	-	-	-	1	1	2	2	1	56.53
b15	447	2	1	3	1400	97.81	24.61	15.46	0	-	-	-	-	-	-	-	-	-	-	61.57
b15	447	3	0	2	1400	97.85	24.61	15.46	20	-	26.15	-	-	1	1	1	4	3	14	106.89
b15	447	3	1	2	1400	97.85	24.61	15.45	0	-	-	-	-	-	-	-	-	-	-	113.68
b15	447	4	0	3	1400	97.83	24.61	15.45	14	-	25.96	-	-	-	2	2	1	5	3	179.64
b15	447	4	1	2	1400	97.83	24.61	15.45	0	-	-	-	-	-	-	-	-	-	-	184.90
systemcaes	670	0	0	0	719	98.64	25.86	22.61	331	31.33	31.39	7381	13066	17562	21031	23791	26383	28111	29537	23.88
systemcaes	670	1	1	6	719	98.66	25.49	21.40	0	-	-	-	-	-	-	-	-	-	-	214.91
systemcaes	670	2	0	1	719	98.63	25.49	21.40	72	-	26.64	-	2	3	12	11	14	32	26	240.95
systemcaes	670	2	1	3	719	98.63	25.49	21.34	0	-	-	-	-	-	-	-	-	-	-	272.56
systemcaes	670	3	0	2	719	98.64	25.49	21.34	86	-	29.32	-	1	5	7	16	17	28	33	318.28
systemcaes	670	3	1	6	719	98.64	25.49	21.26	0	-	-	-	-	-	-	-	-	-	-	359.83
systemcaes	670	4	0	3	719	98.64	25.49	21.26	65	-	27.65	-	-	5	8	9	14	27	33	432.34
systemcaes	670	4	1	6	719	98.64	25.44	21.20	0	-	-	-	-	-	-	-	-	-	-	464.18
s38417	1636	0	0	0	2357	98.12	25.90	23.35	370	27.10	27.33	6191	11597	15996	19985	23228	26043	28491	31034	39.22
s38417	1636	1	1	1	2357	98.12	25.70	23.24	23	26.03	26.10	13	21	45	48	47	76	72	98	159.52
s38417	1636	1	1	2	2357	98.12	25.70	23.24	0	-	-	-	-	-	-	-	-	-	-	167.71

After adding multiple transition faults to F, the first subprocedure computes the faulty switching activity $swa(t, f)$ for every fault $f \in F$ and test $t \in T$. This requires fault simulation without fault dropping of F under T. A fault $f \in F$ creates excessive faulty switching activity under a test $t \in T$ if $swa(t, f) > swa_{max}$. If $swa(t, f) > swa_{max}$ for any test $t \in T$, the fault f is marked as causing excessive faulty switching activity by assigning $\epsilon(f) = 1$. Otherwise, $\epsilon(f) = 0$.

An arbitrary iteration of the first subprocedure starts by removing from F every multiple transition fault f for which $\epsilon(f) = 0$. Such a fault does not cause excessive faulty switching activity under T. The subprocedure prefers to replace this fault with a different one that potentially creates excessive faulty switching activity. The removal of faults from F ensures that the set does not grow to become unmanageable.

For a parameter N_1, the first subprocedure performs N_1 iterations where it updates the set of faults F to prefer faults with excessive faulty switching activity. With the removal of faults from F, the number of faults in F remains close to $2LM$.

C. Numbers of Iterations

With a higher number of iterations, or a higher value of N_1, the first subprocedure finds more multiple transition faults with excessive faulty switching activity. However, the benefit of identifying more and more faults diminishes. This is because the second subprocedure modifies the tests to decrease their fault-free switching activity so as to create a safety margin for excessive faulty switching activity. The safety margin for a test depends on the available faults, but the test accommodates

TABLE III
EXPERIMENTAL RESULTS ($26\% \leq \text{swa}_{\max} < 35\%$)

							ff swa		fy swa										
circuit	sv	it0	sub	it1	tests	f.c.(%)	max(%)	ave(%)	tests	swa1(%)	swa2(%)	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	ntime
s9234	228	0	0	0	633	95.47	26.58	24.79	340	29.36	29.82	2533	4662	6276	7661	8822	9721	10626	11306	15.57
s9234	228	1	1	3	633	95.48	26.15	23.97	0	-	-	-	-	-	-	-	-	-	-	149.47
s9234	228	2	0	1	633	95.47	26.15	23.97	57	-	27.33	-	1	3	3	4	11	10	33	165.60
s9234	228	2	1	2	633	95.48	26.15	23.91	0	-	-	-	-	-	-	-	-	-	-	190.01
s9234	228	3	0	2	633	95.46	26.15	23.91	52	-	28.83	-	-	2	3	5	17	14	20	221.75
s9234	228	3	1	3	633	95.46	26.15	23.84	0	-	-	-	-	-	-	-	-	-	-	245.38
s9234	228	4	0	3	633	95.50	26.15	23.84	46	-	27.64	-	1	2	7	6	9	9	13	289.62
s9234	228	4	1	2	633	95.50	26.15	23.79	0	-	-	-	-	-	-	-	-	-	-	306.70
usb_phy	98	0	0	0	141	98.96	27.71	25.09	45	30.36	31.10	183	357	504	580	708	742	861	886	23.38
usb_phy	98	1	1	2	141	98.96	27.13	24.44	0	-	-	-	-	-	-	-	-	-	-	40.56
usb_phy	98	2	0	1	141	98.93	27.13	24.44	9	-	28.29	-	1	1	4	2	5	2	4	63.91
usb_phy	98	2	1	2	141	98.93	27.13	24.40	0	-	-	-	-	-	-	-	-	-	-	68.88
usb_phy	98	3	0	2	141	98.92	27.13	24.40	10	-	28.12	-	1	1	-	1	4	2	3	115.59
usb_phy	98	3	1	1	141	98.92	27.05	24.31	0	-	-	-	-	-	-	-	-	-	-	119.57
usb_phy	98	4	0	3	141	98.90	27.05	24.31	7	-	28.29	-	-	-	2	-	1	3	2	189.65
usb_phy	98	4	1	1	141	98.90	26.88	24.28	0	-	-	-	-	-	-	-	-	-	-	192.75
spi	229	0	0	0	779	99.46	27.96	21.95	198	31.02	32.24	1170	2246	3090	3792	4385	5102	5409	5999	26.36
spi	229	1	1	4	779	99.46	26.82	21.44	0	-	-	-	-	-	-	-	-	-	-	58.02
spi	229	2	0	1	779	99.46	26.82	21.44	47	-	29.38	-	1	4	3	8	9	10	15	83.90
spi	229	2	1	2	779	99.46	26.82	21.37	0	-	-	-	-	-	-	-	-	-	-	91.86
spi	229	3	0	2	779	99.45	26.82	21.37	80	-	30.48	-	4	4	3	13	14	16	23	143.97
spi	229	3	1	4	779	99.45	26.82	21.26	0	-	-	-	-	-	-	-	-	-	-	160.46
spi	229	4	0	3	779	99.46	26.82	21.26	49	-	31.50	-	2	1	3	11	8	10	14	240.19
spi	229	4	1	4	779	99.46	26.67	21.18	0	-	-	-	-	-	-	-	-	-	-	251.70
s38584	1452	0	0	0	2013	94.65	29.66	17.13	14	30.39	30.39	455	814	1164	1371	1560	1840	2009	2080	13.90
s38584	1452	1	1	1	2013	94.65	29.58	17.13	0	-	-	-	-	-	-	-	-	-	-	19.77
s38584	1452	2	0	1	2013	94.69	29.58	17.13	1	-	29.70	-	-	-	-	-	1	3	-	33.04
s38584	1452	2	1	1	2013	94.69	29.58	17.13	0	-	-	-	-	-	-	-	-	-	-	34.36
s38584	1452	3	0	2	2013	94.68	29.58	17.13	3	-	29.70	-	-	-	1	1	3	1	1	60.91
s38584	1452	3	1	1	2013	94.68	29.57	17.13	0	-	-	-	-	-	-	-	-	-	-	62.88
sasc	117	0	0	0	113	99.29	30.16	27.20	66	33.03	34.92	187	358	450	612	696	719	763	882	27.04
sasc	117	1	1	2	113	99.29	29.44	25.52	0	-	-	-	-	-	-	-	-	-	-	52.87
sasc	117	2	0	1	113	99.27	29.44	25.52	7	-	30.55	-	-	-	-	-	2	2	2	79.74
sasc	117	2	1	2	113	99.27	29.44	25.48	0	-	-	-	-	-	-	-	-	-	-	84.57
sasc	117	3	0	2	113	99.24	29.44	25.48	12	-	31.40	-	-	-	1	4	5	3	1	138.28
sasc	117	3	1	1	113	99.24	29.44	25.32	0	-	-	-	-	-	-	-	-	-	-	143.53
sasc	117	4	0	3	113	99.26	29.44	25.32	15	-	32.70	-	-	1	2	-	2	5	4	224.03
sasc	117	4	1	2	113	99.26	29.44	25.15	0	-	-	-	-	-	-	-	-	-	-	231.64
i2c	128	0	0	0	129	98.49	31.19	25.94	34	34.50	35.34	183	327	448	619	698	833	907	980	23.79
i2c	128	1	1	2	129	98.49	30.49	25.21	0	-	-	-	-	-	-	-	-	-	-	39.83
i2c	128	2	0	1	129	98.46	30.49	25.21	10	-	32.45	-	-	1	2	1	2	3	4	63.55
i2c	128	2	1	1	129	98.46	30.49	25.14	0	-	-	-	-	-	-	-	-	-	-	68.71
i2c	128	3	0	2	129	98.51	30.49	25.14	10	-	32.59	-	-	1	3	2	2	4	-	116.37
i2c	128	3	1	1	129	98.51	30.02	25.05	0	-	-	-	-	-	-	-	-	-	-	121.37
i2c	128	4	0	3	129	98.48	30.02	25.05	8	-	32.26	-	-	-	1	3	-	-	3	189.99
i2c	128	4	1	2	129	98.48	30.02	25.02	0	-	-	-	-	-	-	-	-	-	-	196.12
b20	494	0	0	0	820	97.80	33.14	23.93	43	35.16	35.17	1484	2781	4098	5112	6256	7271	8077	9088	10.63
b20	494	1	1	2	820	97.80	32.61	23.86	0	-	-	-	-	-	-	-	-	-	-	22.79
b20	494	2	0	1	820	97.82	32.61	23.86	9	-	34.47	-	-	-	-	3	1	2	4	36.54
b20	494	2	1	2	820	97.82	32.61	23.85	0	-	-	-	-	-	-	-	-	-	-	40.80
b20	494	3	0	2	820	97.78	32.61	23.85	16	-	34.05	-	-	1	1	2	4	7	7	67.49
b20	494	3	1	2	820	97.78	32.61	23.84	0	-	-	-	-	-	-	-	-	-	-	72.64
b20	494	4	0	3	820	97.81	32.61	23.84	13	-	34.15	-	-	-	-	1	7	4	5	103.78
b20	494	4	1	3	820	97.81	32.41	23.82	0	-	-	-	-	-	-	-	-	-	-	109.18

other faults with similar faulty switching activity. Thus, the modification is effective in eliminating excessive faulty switching activity even for faults that were not considered directly. As more faults are included in F, they become less likely to affect the modification of the test set.

To accommodate both arguments for a low and high value of N_1, N_1 starts from zero, and increased by one with every additional iteration of the procedure. Thus, as the procedure performs more iterations, the first subprocedure also performs more iterations where it attempts to identify additional multiple transition faults with excessive faulty switching activity. Such faults become more difficult to find as the second subprocedure modifies the test set further. Therefore, increasing the number of iterations for selecting multiple transition faults is appropriate.

For the experiments reported in the next section, the procedure from Fig. 3 performs $N_0 = 4$ iterations. Additional iterations are possible, but the benefit from additional iterations diminishes.

The computational effort of the procedure is captured by the normalized runtime, which is the cumulative runtime divided by the runtime for fault simulation with fault dropping of the initial set of transition faults under the initial test set. This is appropriate since the procedure is based on fault simulation.

V. EXPERIMENTAL RESULTS

The procedure from Fig. 3 is applied to benchmark circuits with the following parameter values.

The upper bound on the multiplicity of a transition fault is $M = 8$.

The initial test set T is a low-power broadside test set for single transition faults from [20]. The maximum fault-free

TABLE IV
EXPERIMENTAL RESULTS ($swa_{max} \geq 35\%$)

							ff swa		fy swa											
circuit	sv	it0	sub	it1	tests	f.c.(%)	max(%)	ave(%)	tests	swa1(%)	swa2(%)	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	ntime
aes_core	530	0	0	0	834	99.99	38.24	32.96	1	38.43	38.50	2287	4197	5777	7570	8997	10189	11004	11883	44.22
aes_core	530	1	1	1	834	99.99	37.72	32.96	0	-	-	-	-	-	-	-	-	-	-	45.19
b14	247	0	0	0	451	97.18	41.20	28.04	6	42.06	44.02	143	295	397	498	562	706	700	777	8.63
b14	247	1	1	1	451	97.18	40.19	28.03	0	-	-	-	-	-	-	-	-	-	-	10.90
b14	247	2	0	1	451	97.21	40.19	28.03	1	-	41.37	-	-	-	-	-	1	2	1	22.28
b14	247	2	1	1	451	97.21	40.19	28.03	0	-	-	-	-	-	-	-	-	-	-	23.91
b14	247	3	0	2	451	97.16	40.19	28.03	4	-	41.99	-	-	-	-	1	-	1	2	40.71
b14	247	3	1	1	451	97.16	39.97	28.03	0	-	-	-	-	-	-	-	-	-	-	42.36
b14	247	4	0	3	451	97.19	39.97	28.03	3	-	41.43	-	-	1	1	1	-	1	1	67.41
b14	247	4	1	1	451	97.19	39.91	28.02	0	-	-	-	-	-	-	-	-	-	-	69.18
des_area	128	0	0	0	281	100.00	42.99	34.71	20	44.78	46.20	440	807	1169	1432	1680	1860	2084	2175	26.52
des_area	128	1	1	5	281	100.00	41.44	34.50	0	-	-	-	-	-	-	-	-	-	-	35.86
des_area	128	2	0	1	281	100.00	41.44	34.50	8	-	44.17	-	1	-	2	2	2	1	2	65.20
des_area	128	2	1	9	281	100.00	41.44	34.42	0	-	-	-	-	-	-	-	-	-	-	78.93
des_area	128	3	0	2	281	100.00	41.44	34.42	5	-	43.95	-	-	-	1	-	1	3	1	151.26
des_area	128	3	1	3	281	100.00	41.44	34.40	0	-	-	-	-	-	-	-	-	-	-	156.04
des_area	128	4	0	3	281	100.00	41.44	34.40	11	-	45.37	-	-	1	3	1	2	6	5	270.26
des_area	128	4	1	7	281	100.00	41.44	34.30	0	-	-	-	-	-	-	-	-	-	-	284.49
b07	51	0	0	0	89	96.67	43.90	25.23	1	46.59	47.00	33	66	84	93	120	123	124	146	12.00
b07	51	1	1	1	89	96.67	42.67	25.18	0	-	-	-	-	-	-	-	-	-	-	13.30
b07	51	2	0	1	89	96.66	42.67	25.18	1	-	44.11	-	-	-	1	-	-	-	-	25.35
b07	51	2	1	1	89	96.66	42.67	25.17	0	-	-	-	-	-	-	-	-	-	-	26.65
b07	51	4	0	2	89	96.68	42.67	25.17	2	-	45.66	-	-	-	-	-	-	1	-	74.89
b07	51	4	1	1	89	96.69	42.67	25.12	0	-	-	-	-	-	-	-	-	-	-	88.53
s5378	179	0	0	0	368	98.91	46.19	39.68	18	47.12	47.46	317	584	860	968	1088	1193	1308	1320	56.37
s5378	179	1	1	2	368	98.91	45.40	39.63	0	-	-	-	-	-	-	-	-	-	-	66.08
s5378	179	2	0	1	368	98.92	45.40	39.63	4	-	46.31	-	-	-	-	-	-	2	2	122.35
s5378	179	2	1	1	368	98.92	45.14	39.63	0	-	-	-	-	-	-	-	-	-	-	125.04
s5378	179	3	0	2	368	98.92	45.14	39.63	3	-	46.61	-	-	-	-	1	-	-	2	237.35
s5378	179	3	1	1	368	98.92	45.12	39.62	0	-	-	-	-	-	-	-	-	-	-	239.44
s5378	179	4	0	3	368	98.93	45.12	39.62	6	-	46.25	-	-	-	1	-	1	2	3	404.07
s5378	179	4	1	1	368	98.93	45.12	39.61	0	-	-	-	-	-	-	-	-	-	-	407.28
systemcdes	190	0	0	0	181	99.95	49.54	42.83	9	50.92	51.53	332	579	860	998	1130	1326	1388	1454	49.74
systemcdes	190	1	1	3	181	99.95	47.42	42.74	0	-	-	-	-	-	-	-	-	-	-	56.71
systemcdes	190	2	0	1	181	99.95	47.42	42.74	2	-	49.96	-	-	-	-	-	1	1	1	107.41
systemcdes	190	2	1	2	181	99.95	47.36	42.73	0	-	-	-	-	-	-	-	-	-	-	110.48
systemcdes	190	3	0	2	181	99.95	47.36	42.73	3	-	49.94	-	-	-	-	2	-	1	3	208.43
systemcdes	190	3	1	1	181	99.95	47.36	42.72	0	-	-	-	-	-	-	-	-	-	-	210.40
systemcdes	190	4	0	3	181	99.95	47.36	42.72	1	-	49.68	-	-	-	-	1	-	-	-	356.82
systemcdes	190	4	1	1	181	99.95	47.36	42.72	0	-	-	-	-	-	-	-	-	-	-	358.20
s1423	74	0	0	0	113	98.35	54.88	37.24	5	56.50	56.85	68	117	180	200	228	201	262	216	26.15
s1423	74	1	1	1	113	98.35	53.20	37.19	0	-	-	-	-	-	-	-	-	-	-	29.30
s1423	74	2	0	1	113	98.39	53.20	37.19	2	-	55.45	-	-	-	-	-	2	1	1	55.49
s1423	74	2	1	1	113	98.39	53.20	37.17	0	-	-	-	-	-	-	-	-	-	-	57.34
s35932	1728	0	0	0	160	98.16	60.08	47.39	1	60.15	60.19	3159	5429	6972	7890	8577	8837	9019	8914	22.93
s35932	1728	1	1	2	160	98.16	59.94	47.39	0	-	-	-	-	-	-	-	-	-	-	25.24
s35932	1728	3	0	2	160	98.18	59.94	47.39	1	-	60.08	-	-	-	-	-	-	-	1	93.37
s35932	1728	3	1	1	160	98.18	59.91	47.39	0	-	-	-	-	-	-	-	-	-	-	94.51
b04	66	0	0	0	70	98.62	62.35	37.18	1	69.09	69.79	14	29	38	44	43	49	56	47	23.30
b04	66	1	1	1	70	98.62	54.90	37.06	0	-	-	-	-	-	-	-	-	-	-	24.82

switching activity obtained for the test set from [20] is used as the bound swa_{max} on the fault-free and faulty switching activity. The computations in [20] ensure that the maximum fault-free switching activity does not exceed the maximum functional switching activity.

The results are shown in Tables II–IV as follows. The circuits are arranged by increasing value of the switching activity bound swa_{max}. Each table shows circuits with a different range of switching activity bounds. The range is shown in the table caption. For every circuit, the results are first reported for the initial test set T with the initial set of single and multiple transition faults F. Next, for every iteration of the procedure, the results are reported after the last iteration of the subprocedure for selecting multiple faults that increases the number of faults; and after the last iteration of the subprocedure for modifying the test set that reduces the excessive faulty switching activity.

After the circuit name, column *sv* shows the number of state variables. Column *it*0 shows the iteration of the procedure. Column *sub* has a 0 for the subprocedure that selects multiple transition faults, and a 1 for the subprocedure that modifies tests. Column *it*1 shows the iteration of the corresponding subprocedure.

Column *tests* shows the number of tests in the test set. Column $f.c.$ shows the transition fault coverage with respect to the current set of transition faults. The fault coverage may change as different multiple transition faults are included in F.

Column *ff* swa shows the maximum and average fault-free switching activity for the test set.

Column *fy* swa shows the following information for the faulty switching activity. Subcolumn *tests* shows the number of tests with excessive faulty switching activity. Subcolumn swa1 shows the maximum faulty switching activity considering only single faults in F. Subcolumn swa2 shows the maximum faulty switching activity considering all the faults in F.

For $1 \leq \hat{m} \leq 8$, subcolumn $m = \hat{m}$ shows the number of faults in F with excessive faulty switching activity that have multiplicity $\hat{m}$. Column *ntime* shows the normalized runtime of the procedure from Fig. 3.

TABLE V
EXPERIMENTAL RESULTS WITH HAZARDS

							ff swa			fy swa									
circuit	sv	it0	sub	it1	tests	f.c.(%)	max(%)	ave(%)	tests	swa1(%)	swa2(%)	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	ntime
wb_dma	523	0	0	0	747	99.22	27.76	22.91	88	30.90	31.10	145	269	398	502	670	790	900	953	57.26
wb_dma	523	1	1	2	747	99.22	26.49	22.72	0	-	-	-	-	-	-	-	-	-	-	84.45
wb_dma	523	2	0	1	747	99.22	26.49	22.72	23	-	30.16	-	-	2	3	1	3	2	4	148.44
wb_dma	523	2	1	2	747	99.22	26.49	22.69	0	-	-	-	-	-	-	-	-	-	-	157.50
wb_dma	523	3	0	2	747	99.22	26.49	22.69	28	-	29.35	-	1	1	1	3	3	4	5	307.93
wb_dma	523	3	1	2	747	99.22	26.49	22.66	0	-	-	-	-	-	-	-	-	-	-	318.66
wb_dma	523	4	0	3	747	99.22	26.49	22.66	29	-	29.15	-	-	-	4	2	1	5	3	492.54
wb_dma	523	4	1	2	747	99.22	26.49	22.63	0	-	-	-	-	-	-	-	-	-	-	502.78
simple_spi	131	0	0	0	220	98.99	29.23	22.78	37	31.65	31.95	78	149	179	269	330	374	454	445	56.41
simple_spi	131	1	1	2	220	98.99	27.70	22.27	0	-	-	-	-	-	-	-	-	-	-	73.06
simple_spi	131	2	0	1	220	98.98	27.70	22.27	12	-	30.93	-	1	1	1	-	5	2	4	129.36
simple_spi	131	2	1	1	220	98.98	27.70	22.14	0	-	-	-	-	-	-	-	-	-	-	135.10
simple_spi	131	3	0	2	220	99.01	27.70	22.14	5	-	30.07	-	-	-	-	1	-	-	4	247.30
simple_spi	131	3	1	1	220	99.01	27.51	22.08	0	-	-	-	-	-	-	-	-	-	-	250.23
simple_spi	131	4	0	3	220	98.95	27.51	22.08	6	-	29.78	-	-	-	3	2	1	1	3	418.53
simple_spi	131	4	1	2	220	98.95	27.51	22.03	0	-	-	-	-	-	-	-	-	-	-	423.26
usb_phy	98	0	0	0	141	98.96	30.56	26.61	18	32.61	32.71	65	127	169	199	242	266	316	307	70.52
usb_phy	98	1	1	1	141	98.96	29.57	26.42	0	-	-	-	-	-	-	-	-	-	-	83.15
usb_phy	98	2	0	1	141	98.95	29.57	26.42	6	-	30.91	-	-	1	1	2	2	3	2	153.69
usb_phy	98	2	1	1	141	98.95	29.57	26.39	0	-	-	-	-	-	-	-	-	-	-	158.60
usb_phy	98	3	0	2	141	98.92	29.57	26.39	2	-	30.73	-	-	-	-	2	1	-	1	299.31
usb_phy	98	3	1	1	141	98.92	29.49	26.39	0	-	-	-	-	-	-	-	-	-	-	301.62
usb_phy	98	4	0	3	141	98.91	29.49	26.39	4	-	31.31	-	-	-	2	-	-	1	3	512.54
usb_phy	98	4	1	2	141	98.91	29.49	26.38	0	-	-	-	-	-	-	-	-	-	-	517.86
s9234	228	0	0	0	634	95.47	30.80	27.11	41	31.95	32.17	295	569	775	969	1199	1356	1512	1647	36.38
s9234	228	1	1	2	634	95.47	29.82	27.04	0	-	-	-	-	-	-	-	-	-	-	57.16
s9234	228	2	0	1	634	95.44	29.82	27.04	9	-	31.32	-	-	1	-	4	2	1	3	93.67
s9234	228	2	1	1	634	95.44	29.62	27.03	0	-	-	-	-	-	-	-	-	-	-	98.46
s9234	228	3	0	2	634	95.45	29.62	27.03	9	-	31.49	-	-	1	-	2	3	3	4	171.09
s9234	228	3	1	1	634	95.45	29.54	27.02	0	-	-	-	-	-	-	-	-	-	-	175.85
s9234	228	4	0	3	634	95.45	29.54	27.02	6	-	31.01	-	-	-	1	-	1	2	3	284.94
s9234	228	4	1	1	634	95.45	29.54	27.02	0	-	-	-	-	-	-	-	-	-	-	288.46
spi	229	0	0	0	780	99.46	38.18	27.95	20	39.79	39.92	196	378	489	630	746	842	893	999	103.18
spi	229	1	1	3	780	99.46	36.76	27.90	0	-	-	-	-	-	-	-	-	-	-	111.59
spi	229	2	0	1	780	99.46	36.76	27.90	10	-	38.96	-	-	-	-	1	-	5	2	214.76
spi	229	2	1	2	780	99.46	36.76	27.90	0	-	-	-	-	-	-	-	-	-	-	219.24
spi	229	3	0	2	780	99.46	36.76	27.90	14	-	39.35	-	1	1	-	1	2	2	3	425.59
spi	229	3	1	3	780	99.46	36.76	27.87	0	-	-	-	-	-	-	-	-	-	-	432.79
spi	229	4	0	3	780	99.46	36.76	27.87	13	-	41.45	-	-	-	-	1	4	5	1	742.22
spi	229	4	1	2	780	99.46	36.76	27.82	0	-	-	-	-	-	-	-	-	-	-	747.88
sasc	117	0	0	0	113	99.29	36.49	30.29	13	38.38	39.57	64	122	175	220	245	253	279	269	80.51
sasc	117	1	1	1	113	99.29	34.30	30.10	0	-	-	-	-	-	-	-	-	-	-	93.12
sasc	117	3	0	2	113	99.27	34.30	30.10	1	-	36.62	-	-	-	-	-	-	-	1	333.11
sasc	117	3	1	1	113	99.27	34.17	30.10	0	-	-	-	-	-	-	-	-	-	-	335.03
sasc	117	4	0	3	113	99.29	34.17	30.10	7	-	37.48	-	-	-	1	1	1	4	1	574.34
sasc	117	4	1	1	113	99.29	33.86	29.95	0	-	-	-	-	-	-	-	-	-	-	581.48
b14	247	0	0	0	451	97.18	47.30	33.51	6	48.08	49.18	141	283	384	472	534	663	661	725	15.03
b14	247	1	1	1	451	97.18	46.55	33.50	0	-	-	-	-	-	-	-	-	-	-	16.60
b14	247	2	0	1	451	97.17	46.55	33.50	2	-	47.60	-	-	-	-	-	1	-	1	29.96
b14	247	2	1	1	451	97.17	46.55	33.50	0	-	-	-	-	-	-	-	-	-	-	30.98
b14	247	3	0	2	451	97.19	46.55	33.50	2	-	47.43	-	-	-	-	1	-	1	2	55.88
b14	247	3	1	1	451	97.19	46.30	33.49	0	-	-	-	-	-	-	-	-	-	-	56.89
b07	51	0	0	0	89	96.67	48.27	28.92	2	49.85	50.15	31	61	78	80	97	99	98	105	33.93
b07	51	1	1	1	89	96.67	46.28	28.89	0	-	-	-	-	-	-	-	-	-	-	36.15
b07	51	2	0	1	89	96.70	46.28	28.89	2	-	48.86	-	-	-	1	-	-	-	-	70.17
b07	51	2	1	1	89	96.70	46.28	28.86	0	-	-	-	-	-	-	-	-	-	-	72.39
b07	51	3	0	1	89	96.70	46.28	28.86	1	-	48.35	-	-	-	-	-	-	1	-	106.39
b07	51	3	1	1	89	96.79	44.81	28.84	0	-	-	-	-	-	-	-	-	-	-	142.02
b07	51	4	0	2	89	96.68	44.81	28.84	1	-	49.12	-	-	-	-	-	-	1	-	210.03
b07	51	4	1	1	89	96.69	44.81	28.83	0	-	-	-	-	-	-	-	-	-	-	245.63
s5378	179	0	0	0	368	98.91	50.59	42.93	7	51.35	51.59	247	444	577	682	735	853	879	898	75.35
s5378	179	1	1	1	368	98.91	49.92	42.91	0	-	-	-	-	-	-	-	-	-	-	79.59
s5378	179	2	0	1	368	98.92	49.92	42.91	3	-	50.82	-	-	-	-	1	1	1	1	154.87
s5378	179	2	1	1	368	98.92	49.76	42.90	0	-	-	-	-	-	-	-	-	-	-	157.28
s5378	179	4	0	3	368	98.93	49.76	42.90	2	-	50.61	-	-	-	1	-	-	-	1	534.29
s5378	179	4	1	1	368	98.93	49.53	42.90	0	-	-	-	-	-	-	-	-	-	-	536.21
s1423	74	0	0	0	113	98.35	60.37	40.17	1	61.56	61.61	45	76	105	104	119	106	101	73	66.40
s1423	74	1	1	1	113	98.35	58.24	40.15	0	-	-	-	-	-	-	-	-	-	-	68.23

The following points can be seen from Tables II–IV. In the first iteration of the procedure from Fig. 3, before the test set is modified, the procedure finds large numbers of multiple transition faults with excessive faulty switching activity. The difference between the maximum faulty and fault-free switching activity does not decrease with the size of the circuit. Thus, the issue of excessive faulty switching activity occurs independently of the circuit size. To illustrate this point, Fig. 4(a) shows the difference in the maximum switching activity between the faulty and fault-free circuit as a function of the number of state variables for the circuits from Tables II–IV.

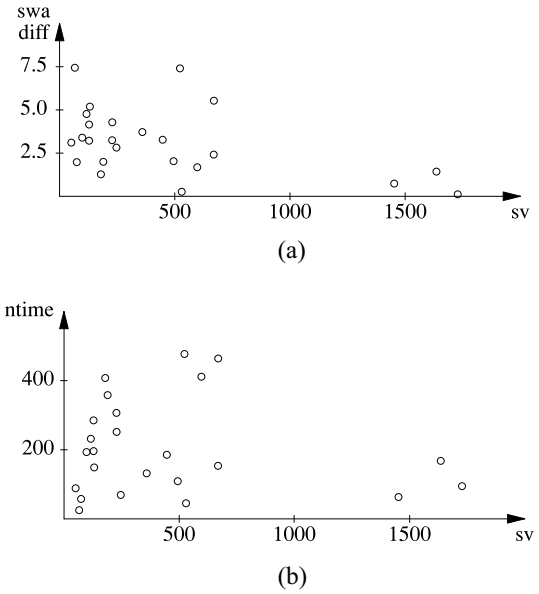

Fig. 4. Scalability of procedure. (a) Difference in maximum switching activity. (b) Normalized runtime.

The number of faults with excessive faulty switching activity increases with the multiplicity m even though the same number of faults is considered for every value of m. This occurs even without iterating through the subprocedure that selects multiple transition faults.

For a circuit with a lower value of the bound on the switching activity, swa_{max}, there is typically a larger number of multiple transition faults with excessive faulty switching activity in the first iteration.

After the subprocedure for modifying the test set is applied in the first iteration, additional iterations find significantly fewer multiple transition faults with excessive faulty switching activity. These faults are typically found among the faults with the higher values of m. However, even these numbers are relatively small, and they decrease with additional iterations. They are typically smaller when swa_{max} is higher for a circuit.

The maximum faulty switching activity for multiple faults is higher than for single faults. This is part of the motivation for considering multiple transition faults in this paper. As additional iterations are performed, the fault-free switching activity decreases, but not significantly. In addition, the maximum faulty switching activity for multiple faults decreases. Thus, the modified test set becomes less susceptible to excessive faulty switching activity even in the presence of multiple transition faults.

The normalized runtime for the initial test set with the initial set of faults is dominated by the computation of the faulty switching activity using fault simulation without fault dropping. As additional iterations are performed, fault simulation without fault dropping is carried out to recompute the faulty switching activity. Overall, the normalized runtime does not increase with the size of the circuit. This is illustrated by Fig. 4(b) that shows the normalized runtime as a function of the number of state variables for the circuits from Tables II–IV. This implies that the procedure scales similar to a fault simulation procedure. The normalized runtime is lower when the bound swa_{max} on the switching activity is higher, and fewer faults cause excessive faulty switching activity.

Finally, to demonstrate that the procedure from Fig. 3 can be applied with different metrics for the power dissipation, the switching activity is replaced with a metric that counts a transition as having a unit contribution, and a potential hazard as having a 0.25 contribution. The simulator used for identifying potential hazards computes a third, intermediate value for every line under a broadside test. The value is unspecified for a transition, and results in an unspecified value when a hazard may occur. The results obtained for several circuits using this metric are shown in Table V. Table V demonstrates the need for the procedure from Fig. 3 when this metric is used.

VI. Conclusion

Excessive switching activity in a faulty circuit can cause a delay fault to escape detection for the same reason that excessive switching activity in the fault-free circuit can result in a fault-free circuit failing a test. This paper observed that the switching activity in a faulty circuit can be higher in the presence of a multiple transition fault with a higher multiplicity. To address the challenge in considering multiple transition faults, this paper described an iterative procedure that has two subprocedures: 1) the first subprocedure selects multiple transition faults iteratively in order to accumulate multiple transition faults with excessive faulty switching activity and 2) the second subprocedure modifies the test set so as to avoid excessive faulty switching activity for the faults found by the first subprocedure. The experimental results for benchmark circuits demonstrated the levels of excessive faulty switching activity that occurs in the presence of multiple transition faults, and the possibility of reducing or even eliminating it after a small number of iterations.

References

[1] J. Saxena *et al.*, "A case study of IR-drop in structured at-speed testing," in *Proc. Int. Test Conf.*, 2003, pp. 1098–1104.
[2] S. Sde-Paz and E. Salomon, "Frequency and power correlation between at-speed scan and functional tests," in *Proc. Int. Test Conf.*, Santa Clara, CA, USA, 2008, pp. 1–9.
[3] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. M. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 12, pp. 1325–1333, Dec. 1998.
[4] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," in *Proc. VLSI Test Symp.*, Montreal, QC, Canada, 2000, pp. 35–40.
[5] K.-J. Lee, T.-C. Huang, and J.-J. Chen, "Peak-power reduction for multiple-scan circuits during test application," in *Proc. Asian Test Symp.*, Taipei, Taiwan, 2000, pp. 453–458.
[6] J. Saxena, K. M. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," in *Proc. Int. Test Conf.*, Baltimore, MD, USA, 2001, pp. 670–677.
[7] S. Kajihara, K. Ishida, and K. Miyase, "Test vector modification for power reduction during scan testing," in *Proc. VLSI Test Symp.*, Monterey, CA, USA, 2002, pp. 160–165.
[8] D. Xiang, S. Gu, J.-G. Sun, and Y.-L. Wu, "A cost-effective scan architecture for scan testing with nonscan test power and test application cost," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2003, pp. 744–747.
[9] W. Li, S. M. Reddy, and I. Pomeranz, "On test generation for transition faults with minimized peak power dissipation," in *Proc. Design Autom. Conf.*, San Diego, CA, USA, 2004, pp. 504–509.

[10] X. Wen *et al.*, "Low-capture-power test generation for scan-based testing," in *Proc. Int. Test Conf.*, 2005, pp. 1019–1028.
[11] V. R. Devanathan, C. P. Ravikumar, and V. Kamakoti, "On power-profiling and pattern generation for power-safe scan tests," in *Proc. Design Autom. Test Europe Conf.*, Nice, France, 2007, pp. 1–6.
[12] J. Lee, S. Narayan, M. Kapralos, and M. Tehranipoor, "Layout-aware, IR-drop tolerant transition fault pattern generation," in *Proc. Design Autom. Test Europe Conf.*, 2008, pp. 1172–1177.
[13] A. Chandra and R. Kapur, "Bounded adjacent fill for low capture power scan testing," in *Proc. VLSI Test Symp.*, San Diego, CA, USA, 2008, pp. 131–138.
[14] P. Girard, N. Nicolici, and X. Wen, Eds., *Power-Aware Testing and Test Strategies for Low Power Devices*. New York, NY, USA: Springer, 2009.
[15] C.-W. Tzeng and S.-Y. Huang, "QC-fill: Quick-and-cool X-filling for multicasting-based scan test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 11, pp. 1756–1766, Nov. 2009.
[16] E. K. Moghaddam, J. Rajski, M. Kassab, and S. M. Reddy, "At-speed scan test with low switching activity," in *Proc. VLSI Test Symp.*, Santa Cruz, CA, USA, 2010, pp. 177–182.
[17] M.-F. Wu *et al.*, "Improved weight assignment for logic switching activity during at-speed test pattern generation," in *Proc. Asia South Pac. Design Autom. Conf.*, Taipei, Taiwan, 2010, pp. 493–498.
[18] K. Miyase, M. Sauer, B. Becker, X. Wen, and S. Kajihara, "Search space reduction for low-power test generation," in *Proc. Asian Test Symp.*, 2013, pp. 171–176.
[19] Y.-H. Li, W.-C. Lien, I.-C. Lin, and K.-J. Lee, "Capture-power-safe test pattern determination for at-speed scan-based testing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 1, pp. 127–138, Jan. 2014.
[20] I. Pomeranz, "Low-power test generation by merging of functional broadside test cubes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 7, pp. 1570–1582, Jul. 2014.
[21] I. Pomeranz, "Skewed-load test cubes based on functional broadside tests for a low-power test set," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 3, pp. 593–597, Mar. 2015.
[22] K. Miyase, M. Sauer, B. Becker, X. Wen, and S. Kajihara, "Identification of high power consuming areas with gate type and logic level information," in *Proc. Eur. Test Symp.*, 2015, pp. 1–6.
[23] M. Filipek *et al.*, "Low-power programmable PRPG with test compression capabilities," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 6, pp. 1063–1076, Jun. 2015.
[24] M. Yoshimura, Y. Takahashi, H. Yamazaki, and T. Hosokawa, "A don't care filling method to reduce capture power based on correlation of FF transitions," in *Proc. Asian Test Symp.*, 2015, pp. 13–18.
[25] I. Pomeranz and S. M. Reddy, "On the switching activity in faulty circuits during test application," in *Proc. Asian Test Symp.*, 2016, pp. 13–18.
[26] I. Pomeranz, "Postprocessing procedure for reducing the faulty switching activity of a low-power test set," in *Proc. Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, 2018, pp. 1–6.

Irith Pomeranz (M'89–SM'96–F'99) received the B.Sc. degree (*summa cum laude*) in computer engineering and the D.Sc. degree from the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel, in 1985 and 1989, respectively.

From 1989 to 1990, she was a Lecturer with the Department of Computer Science, Technion. From 1990 to 2000, she was a Faculty Member with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA. In 2000, she joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. Her current research interests include testing of VLSI circuits, design for testability, and synthesis and design verification.

Dr. Pomeranz was a recipient of the NSF Young Investigator Award in 1993 and the University of Iowa Faculty Scholar Award in 1997. Three of her conference papers won best paper awards, and four other papers were nominated for best paper awards. One of the papers she co-authored was selected by the 2016 International Test Conference as the most significant paper published ten years before. She delivered a keynote speech at the 2006 Asian Test Symposium. She served as an Associate Editor for the *ACM Transactions on Design Automation*, the IEEE TRANSACTIONS ON COMPUTERS, and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. She served as a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS in 1998 special issue on Dependability of Computing Systems, and as the Program Co-Chair of the 1999 Fault-Tolerant Computing Symposium. She served as the Program Chair of the 2004 and 2005 VLSI Test Symposium, and as the General Chair of the 2006 VLSI Test Symposium. She is a Golden Core Member of the IEEE Computer Society.