

Target Faults for Test Compaction based on Multicycle Tests

IRITH POMERANZ, Purdue University

The use of multicycle tests, with several functional capture cycles between scan operations, contributes significantly to the ability to compact a test set. Multicycle tests have the added benefit that they can contribute to the detection of defects with complex behaviors that are not detected by single-cycle or two-cycle tests. To ensure that this benefit is materialized when test compaction is applied to transition faults, this paper suggests to incorporate into the test compaction procedure an additional fault model whose fault coverage increases when multicycle tests are used. To ensure that the computational complexity of test compaction is not increased by a fault model with a large number of faults, or faults with complex behaviors, the added fault model is required to have the same characteristics as the transition fault model. A type of transition faults called unspecified transition faults satisfies these requirements. The paper describes a test compaction procedure for transition faults that incorporates unspecified transition faults, and presents experimental results for benchmark circuits to demonstrate the levels of test compaction and fault coverage that can be achieved.

CCS Concepts: •Hardware →Test-pattern generation and fault simulation;

Additional Key Words and Phrases: Multicycle tests, test compaction, test generation, transition faults.

ACM Reference format:

Irith Pomeranz. 2019. Target Faults for Test Compaction based on Multicycle Tests. *ACM Trans. Des. Autom. Electron. Syst.* 1, 1, Article 1 (January 2019), 13 pages.

DOI: 0000001.0000001

1 INTRODUCTION

Delay fault models capture the behaviors of delay defects in order to enable fault simulation and test generation to be carried out [1]-[17]. Transition faults [1] model the case where a large extra delay is local to a line. The transition fault model is commonly used because the number of transition faults is manageable, being comparable to the number of single stuck-at faults, and the activation and propagation conditions for transition faults are similar to those of single stuck-at faults.

Transition faults are detected by scan tests with two or more functional capture cycles between a scan-in and a scan-out operation (broadside tests). Two or more clock cycles are used for activating the faults and propagating their effects to observable outputs. Two-cycle broadside tests, with two functional capture cycles between scan operations, are sufficient for the detection of transition faults. Multicycle broadside tests, with two or more functional capture cycles between scan operations, provide the following advantages [18]-[26].

(1) They potentially improve the ability of the test set to detect defects with complex behaviors by exercising the circuit at-speed for several functional capture cycles [18].

Author's address: Irith Pomeranz, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, U.S.A. E-mail: pomeranz@ecn.purdue.edu.

The work was supported in part by the National Science Foundation under Grant Number CCF-1714147.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 ACM. 1084-4309/2019/1-ART1 \$15.00

DOI: 0000001.0000001

- (2) They allow test compaction to be achieved. This was originally observed for single stuck-at faults [27]-[28], and can be explained as follows. By increasing the number of functional capture cycles in a test, it is possible to increase the number of faults that the test detects. This allows the number of tests in a test set to be reduced. A reduction in the number of tests implies a reduction in the number of scan operations. This translates into a reduction in the number of clock cycles required for test application, and a reduction in the test data volume.
- (3) Multicycle tests can address features such as multiple clock domains and partial scan [20]-[21].
- (4) They create closer-to-functional operation conditions that are important for avoiding overtesting of delay faults [19].

In existing test compaction procedures that produce multicycle tests for transition faults, the transition fault coverage of the multicycle test set is typically the same as that of a two-cycle test set. A possible formulation of the problem addressed by existing test compaction procedures is the following.

Problem 1: Given a two-cycle broadside test set T_2 for the set F_0 of transition faults, find a multicycle broadside test set T_L that satisfies the following conditions. (1) The number of clock cycles required for application of T_L is as small as possible. (2) T_L detects all the faults from F_0 detected by T_2 .

Within this formulation, transition faults do not allow the test compaction procedure to identify multicycle tests that can potentially improve the defect coverage. As a result, the advantage of multicycle tests in detecting defects with complex behaviors may not be materialized by the resulting test set. To address this issue, the paper suggests to incorporate into the test compaction procedure for transition faults an additional fault model whose achievable fault coverage increases when multicycle tests with more functional capture cycles are used. This will allow the test compaction procedure to identify tests that are likely to increase the defect coverage, and keep them in the test set. Without increasing the defect coverage, the procedure will achieve test compaction. More tests may be needed when the defect coverage is increased. Overall, the procedure will combine the advantages of test compaction and increased defect coverage. The problem addressed by the test compaction procedure described in this paper is the following.

Problem 2: Given a two-cycle broadside test set T_2 for the set F_0 of transition faults, and an additional set F_1 of faults whose achievable fault coverage increases when multicycle tests with more functional capture cycles are used, find a multicycle broadside test set T_L that satisfies the following conditions. (1) The number of clock cycles required for application of T_L is as small as possible. (2) T_L detects all the faults from F_0 detected by T_2 . (3) T_L detects as many faults from F_1 as possible.

A fault model to complement the transition fault model is selected in this paper as follows. The standard transition faults that are typically considered under multicycle tests are associated with an extra delay of a single clock cycle (these faults are included in F_0 of Problems 1 and 2). Transition faults with different durations model delay defects with different durations [4]. These faults have different behaviors than the standard transition faults under multicycle tests, and they are detected by different tests. They can thus be used for enhancing the defect coverage of a multicycle test set. However, the use of transition faults with different durations also increases the number of faults significantly.

To ensure that the computational complexity of test compaction is not increased by a fault model with a large number of faults, or faults with complex behaviors, the added fault model is required to have similar characteristics to the transition fault model. Specifically, the number of faults should be the same, and the fault simulation procedure should have the same computational complexity. A type of transition faults called unspecified transition faults provides such a fault model [5]. An unspecified transition fault captures the behaviors of transition faults with different durations. The

unspecified transition fault model is thus suitable as a target for increasing the defect coverage of a multicycle test set.

The paper describes a test compaction procedure where unspecified transition faults complement the use of standard transition faults during the generation of multicycle tests. Experimental results for benchmark circuits demonstrate that the test set is compacted even in cases where the fault coverage is increased. To demonstrate the importance of targeting an additional fault model as well as the test compaction capabilities of the procedure, it is divided into three phases. Phase 1 achieves test compaction without considering unspecified transition faults. Phase 2 adds tests for unspecified transition faults without attempting to compact the test set, demonstrating that many of these faults are not detected accidentally by multicycle tests for standard transition faults. Phase 3 applies the test compaction procedure using both fault models in order to compact the test set further. Once the effectiveness of the procedure is established, only phase 3 needs to be applied.

Unspecified transition faults are used in this paper as an example of a fault model whose fault coverage increases when multicycle tests are used, and has the same characteristics as the commonly used transition fault model. Other fault models with similar characteristics can be used instead of unspecified transition faults to guide the test compaction procedure.

The test compaction procedure described in this paper is simulation based in order to avoid the need for sequential test generation to generate multicycle tests. The procedure creates a pool of multicycle tests based on a two-cycle test set, and selects a subset using the criteria included in Problem 2. This limits its computational complexity at the cost of giving up optimality. Other test compaction procedures can be used instead.

The paper is organized as follows. Unspecified transition faults are reviewed in Section 2. Section 3 describes the test compaction procedure, and its partition into three phases. Experimental results are presented in Section 4.

2 UNSPECIFIED TRANSITION FAULTS

Delay faults are used for modeling delay defects that affect the timing of a circuit. Such defects cause the circuit to fail at its designated speed of operation. Transition faults are a type of delay faults where the extra delay of a line is assumed to be large enough to cause the circuit to fail independent of the delays of the paths through which it is propagated. The transition fault denoted by $g : a \rightarrow a'$ delays the $a \rightarrow a'$ transition on line g . In the presence of the fault, the transition does not occur within the clock period of the circuit. A two-cycle test for the fault assigns $g = a$ in the first clock cycle and $g = a'$ in the second clock cycle in order to activate the fault. In the faulty circuit, the transition does not occur, and $g = a$ in both clock cycles. This creates the fault effect $g = a'/a$ in the second clock cycle. The second clock cycle of the test needs to propagate the difference between the fault-free and faulty values of g to an observable output.

A multicycle test has two or more functional capture cycles between its scan operations. When a sequence of functional capture cycles is simulated at-speed, a transition fault may have a different effect on the output response of the circuit depending on the duration of the extra delay. While the effects of a transition fault with an extra delay of a single clock cycle disappear after one clock cycle, a transition fault with an extra delay of two clock cycles can affect the circuit for two clock cycles. To address this issue, each transition fault is considered in [4] with different durations of the extra delay on the faulty line. The duration is measured in numbers of clock cycles, and each number defines a different fault. Each one of the faults needs to be simulated separately as an independent fault. This increases the computational complexity for the consideration of transition faults.

To keep the number of transition faults from growing, only transition faults with a delay of a single clock cycle are typically considered. These faults tend to be the most difficult to detect,

making it less important to consider faults with larger delays. A transition fault with an extra delay of a single clock cycle is referred to as a standard transition fault.

Unspecified transition faults are defined in [5] as an alternative to the approach from [4]. The unspecified transition fault model captures all the possible durations of a delayed transition in a single fault by introducing unspecified values into the faulty circuit when fault effects may occur. The unspecified values indicate that a fault effect may occur depending on the duration of the fault. Fault detection is assumed to occur when an unspecified value reaches an observable output. To increase the confidence that the fault is detected, several unspecified values are required to reach an observable output before a fault is considered to be detected in [5].

The difference between standard and unspecified transition faults is related to their activation. This issue is discussed next.

For a standard transition fault $g : a \rightarrow a'$ under a two-cycle test, having $g = a$ in the first clock cycle, and $g = a'$ in the second clock cycle, causes the transition fault to be activated. This results in the value $g = a$ in the faulty circuit under the second cycle.

A pessimistic unspecified transition fault is activated by two consecutive clock cycles such that $g = a$ in the first clock cycle, and $g = a'$ in the second clock cycle. However, in this case, an unspecified (x) value is assigned to g in the faulty circuit under the second cycle. The unspecified value represents the uncertainty regarding the duration of the fault, which determines whether or not the fault will persist in the next clock cycle.

Using the definition from [5], an optimistic unspecified transition fault $g : a \rightarrow a'$ is activated if the value of line g in the first clock cycle is a or x , and the value of g in the second clock cycle is a' . This causes an unspecified value to be assigned to g in the faulty circuit under the second clock cycle. The fault is referred to as optimistic because of the assumption that the conditions for the activation of g will be created even with an unspecified value on g . This causes more unspecified values to propagate through the circuit. Since unspecified values do not cancel each other, this increases the likelihood that the fault will be detected.

An even more optimistic definition of an unspecified transition fault allows g to assume an unspecified value in the second clock cycle as well. This is the definition used in this paper. Thus, an optimistic unspecified transition fault $g : a \rightarrow a'$ is activated if the value of line g in the first clock cycle is a or x , and the value of g in the second clock cycle is a' or x .

With this definition, the number of optimistic unspecified transition faults is the same as the number of standard transition faults, and the fault simulation procedures for the two fault models have the same computational complexity.

The following example illustrates the detection of an optimistic unspecified transition fault using a multicycle test with three functional capture cycles. The example is based on benchmark circuit s298. The circuit has 14 state variables, three primary inputs, and six primary outputs. Figure 1 shows the functional clock cycles of a three-cycle broadside test, and the values it assigns in the presence of a falling optimistic unspecified transition fault. In the second clock cycle, the value 0 (shown in parentheses) is computed for the faulty line. The fault is activated, causing the value of the faulty line to be changed to x . In the third clock cycle, the value 0 is computed for the faulty line again. The fault is activated again, causing the value of the faulty line to be changed to x . This results in an x value reaching the final state of the test, which is scanned out.

3 TEST COMPACTION PROCEDURE

This section describes a test compaction procedure that produces multicycle tests for transition faults. The complete procedure is described first, followed by its partition into three phases that demonstrate its various capabilities.

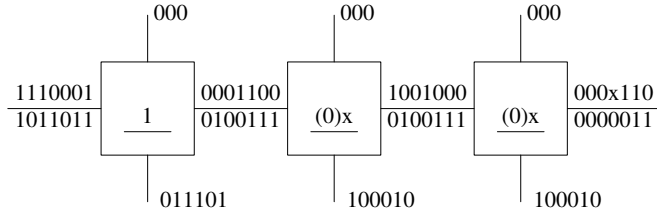


Fig. 1. Three-cycle broadside test

3.1 Overview

The procedure considers a set of target faults F that consists of standard as well as optimistic unspecified transition faults. For every line g and value a , F contains the standard as well as the optimistic unspecified transition fault $g : a \rightarrow a'$.

An l_i -cycle broadside test is denoted by $t_i = \langle s_i, v_i, l_i \rangle$, where s_i is the scan-in state, v_i is a primary input vector, and l_i is the number of functional capture cycles of the test. The primary input vector v_i is held constant for the duration of the test.

The test compaction procedure accepts a compact two-cycle broadside test set for standard transition faults. The test set is denoted by T_2 . The test set T_2 detects all or most of the detectable standard transition faults from F , and typically the same number of unspecified transition faults. It does not detect all the detectable optimistic unspecified transition faults from F , since these faults were not targeted during the generation of T_2 , and some of them may only be detected by multicycle tests with more than two functional capture cycles.

The procedure produces a series of test sets, $T_3, T_4, \dots$. A test $t_i \in T_L$ is an l_i -cycle broadside test, for $2 \leq l_i \leq L$. The goal of producing T_L , for $L \geq 3$, is to compact the test set, while also detecting additional unspecified transition faults from F . It should be noted that these two goals can be conflicting, and an increased unspecified transition fault coverage may require an increased number of tests. The use of multicycle tests provides test compaction even in this case since it allows each test to detect more faults than a two-cycle test. The increase in the number of tests is accepted to support the increased fault coverage.

For an arbitrary value of $L \geq 3$, the test set T_L is constructed as illustrated by Figure 2. Phase 3 is the one relevant to the discussion in this section. The other phases will be explained later.

Initially, $T_L = T_{L-1}$. The procedure adds L -cycle broadside tests to T_L as described below. The goal is to create a pool of L -cycle broadside tests that are likely to be effective for two purposes: (1) detecting faults from F that are not detected by T_{L-1} , and (2) contributing to test compaction.

After the pool is constructed, a set covering procedure is used for selecting a minimal subset of T_L that is sufficient for detecting all the faults from F that the pool detects. The inclusion of T_{L-1} in T_L ensures that a fault, whose detection requires an l_i -cycle broadside test $t_i \in T_{L-1}$, can be detected by T_L .

Two procedures that are used for constructing the pool are described next, followed by the set covering procedure.

3.2 Constructing the Pool

After assigning $T_L = T_{L-1}$, the procedure adds L -cycle broadside tests to T_L as follows.

Experimental results indicate that the tests in T_2 yield multicycle tests that are effective in detecting standard as well as optimistic unspecified transition faults simply by increasing their numbers of functional clock cycles, and using the same scan-in states and primary input vectors.

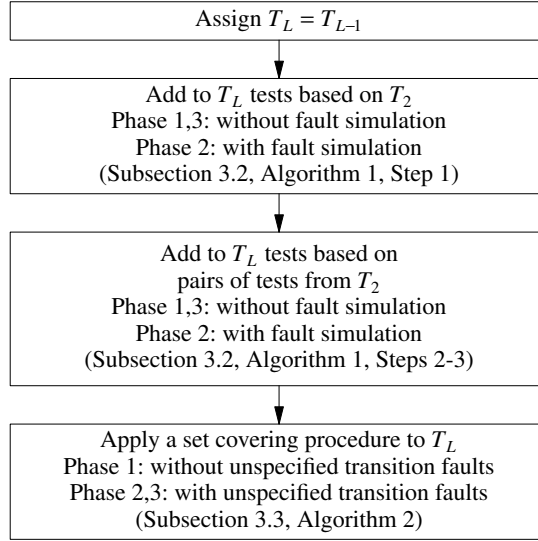


Fig. 2. Test compaction procedure

Thus, considering a test $t_i^2 = \langle s_i, v_i, 2 \rangle \in T_2$, there are many cases where a test $t_i^L = \langle s_i, v_i, L \rangle$, with $L > 2$, is effective in detecting standard and optimistic unspecified transition faults. This is explained by the fact that the tests in T_2 create the activation and propagation conditions for standard transition faults. By keeping the same scan-in state and primary input vector, and adding functional capture cycles, activation and propagation conditions are created for both standard and optimistic unspecified transition faults. Based on this observation, the procedure adds tests to T_L as follows.

For every test $t_i^2 = \langle s_i, v_i, 2 \rangle \in T_2$, the procedure adds to T_L the L -cycle test $t_i^L = \langle s_i, v_i, L \rangle$.

To allow more variation in the tests that are considered for T_L , the procedure combines values of scan-in states and primary input vectors from different tests in T_2 . Considering $t_i = \langle s_i, v_i, 2 \rangle \in T_2$ and $t_j = \langle s_j, v_j, 2 \rangle \in T_2$ such that $i \neq j$, the L -cycle test obtained by combining t_i and t_j is denoted by $t_{i,j}^L = \langle s_{i,j}, v_{i,j}, L \rangle$. For every bit of $s_{i,j}$ and $v_{i,j}$, the procedure selects one of t_i or t_j randomly, and copies the value of the bit from the selected test. By combining values from two tests, the procedure combines the activation and propagation conditions they create. In many cases, this allows standard as well as optimistic unspecified transition faults to be detected.

The number of pairs of tests that are available for defining multicycle tests is $|T_2|(|T_2| - 1)$, and every pair can result in several different tests depending on how the bits of the pair are combined. To avoid creating a pool with an excessive number of tests, the procedure uses the following approach.

The Hamming distance between $t_i \in T_2$ and $t_j \in T_2$ is the number of bits where s_i and s_j , and v_i and v_j , differ. A pair of tests with a larger Hamming distance allows more variation in the combined test. Experimental results indicate that this variation is important for the detection of optimistic unspecified transition faults. To focus on the pairs of tests with the largest Hamming distances, the procedure includes in a set that is denoted by P_0 the pairs of tests with the largest Hamming distances. The number of pairs that the procedure includes in P_0 is $N_{P0}|T_2|$, for a constant N_{P0} .

To define multicycle tests, the procedure selects pairs of tests from P_0 randomly. The number of pairs that the procedure selects is $N_{P1}|T_2|$, for a constant $N_{P1} < N_{P0}$. The procedure combines every pair once to produce one multicycle test for every pair.

The algorithm for constructing the pool is summarized next.

Algorithm 1: Constructing the pool

- (1) For every test $t_i^2 = \langle s_i, v_i, 2 \rangle \in T_2$, add to T_L the L -cycle test $t_i^L = \langle s_i, v_i, L \rangle$.
- (2) Include in a set P_0 the $N_{P0}|T_2|$ pairs of tests from T_2 with the largest Hamming distances.
- (3) Repeat $N_{P1}|T_2|$ times:
 - (a) Select a pair of tests (t_i, t_j) from P_0 randomly.
 - (b) Define an L -cycle test $t_{i,j}^L = \langle s_{i,j}, v_{i,j}, L \rangle$ as follows. For every bit of $s_{i,j}$ and $v_{i,j}$, select one of t_i or t_j randomly, and copy the value of the bit from the selected test.
 - (c) Add $t_{i,j}^L$ to T_L .

The procedure uses $N_{P0} = 8$ and $N_{P1} = 4$. This ensures that the number of tests in T_L is increased by at most $4|T_2|$, while providing $8|T_2|$ candidate pairs of tests to create variation in the test set for different values of L . Experimental results show that these parameter values are effective at balancing the computational effort, the level of test compaction, and the unspecified transition fault coverage.

It should be noted that L -cycle tests are added to T_L without fault simulation. This creates a large pool of L -cycle tests, from which effective ones can be selected.

3.3 Set Covering Procedure

The set covering procedure described in this section selects a minimal subset of the tests in T_L that are needed for detecting faults from F .

Initially, all the tests in T_L are marked as unselected, and all the faults in F are marked as uncovered.

The procedure performs N -detection fault simulation of F under T_L , for a constant $N > 1$. During N -detection fault simulation, a fault is dropped only after it is detected by N different tests. Thus, for every fault $f_j \in F$ that is detected by T_L , N -detection fault simulation yields up to N tests from T_L that detect f_j . The subset of tests that detect f_j is denoted by $D_j \subseteq T_L$. For the experiments in this paper, $N = 16$ is used for ensuring that a sufficiently large number of tests is available for every fault.

The procedure considers the uncovered faults from F , which are detected by T_L , one at a time in an increasing order of the number of tests in D_j . This order is motivated by the fact that a fault with a smaller subset D_j is less likely to be detected by tests that detect other faults, and it is more likely that a test will have to be selected for it directly out of D_j .

When a fault $f_j \in F$ is considered, the procedure selects one of the tests from D_j to cover the fault. For a test $t_i \in D_j$, let c_i be the number of uncovered faults that the test detects. This is the number of uncovered faults $f_k \in F$ such that $t_i \in D_k$. Let l_i be the number of functional capture cycles in t_i . Of all the tests in D_j , the procedure selects the test t_i for which c_i is the largest, and l_i is the smallest, in this order of importance.

The procedure marks t_i as selected. It also marks as covered every uncovered fault $f_k \in F$ such that $t_i \in D_k$. With the largest number of covered faults, t_i is expected to be the most effective in forming a compact test set. With the smallest number of clock cycles l_i , the procedure avoids selecting tests with increased numbers of functional capture cycles unless this is important for the detection of the faults in F .

After all the faults in F are covered, the procedure keeps in T_L only the tests that are marked as selected. The procedure compares T_L with T_{L-1} based on two parameters. The number of clock

cycles required for the application of T_L is denoted by Σ_L . This includes scan shift cycles for scan operations, and functional capture cycles. The number of clock cycles is used for measuring the level of test compaction, and not the number of tests, since different tests may have different numbers of functional capture cycles. The second parameter is the number of standard and optimistic unspecified transition faults that the test set detects. For T_L , this number is denoted by Δ_L .

The procedure considers T_L to be improved compared with T_{L-1} if $D_L > D_{L-1}$ or $\Sigma_L < \Sigma_{L-1}$. If T_L is not improved compared with T_{L-1} , the procedure prefers T_{L-1} over T_L , and assigns $T_L = T_{L-1}$.

The set covering algorithm is summarized next.

Algorithm 2: Set covering

- (1) Mark all the tests in T_L as unselected. Mark all the faults in F as uncovered.
- (2) Perform N -detection fault simulation of F under T_L . For every fault $f_j \in F$, store the tests that detect f_j in a set D_j .
- (3) If all the faults in F are either covered or have $D_j = \emptyset$, remove all the unselected tests from T_L , and stop.
- (4) Select an uncovered fault $f_j \in F$ with the smallest subset D_j such that $D_j \neq \emptyset$.
- (5) For every test $t_i \in D_j$, let c_i be the number of uncovered faults $f_k \in F$ such that $t_i \in D_k$.
- (6) Select the test t_i for which c_i is the largest, and l_i is the smallest, in this order of importance.
- (7) Mark t_i as selected. Mark every uncovered fault $f_k \in F$ such that $t_i \in D_k$ as covered.
- (8) Go to Step 3.

3.4 Three-Phase Procedure

To demonstrate the effectiveness of the test compaction procedure, it is important to demonstrate the following points. (1) There are significant numbers of optimistic unspecified transition faults that are not detected accidentally by multicycle tests for standard transition faults, and it is important to target these faults directly during test compaction. (2) The use of multicycle tests contributes to test compaction even when the fault coverage is increased.

These two goals are achieved together by the complete procedure. As a result, the increase in the coverage of unspecified transition faults may hide the ability of the procedure to achieve test compaction, and vice versa. To demonstrate both of these points, the test compaction procedure is applied in three phases. Each phase applies the procedure differently. Figure 2 shows how the procedure is modified in every phase.

Phase 1 achieves test compaction for standard transition faults without considering optimistic unspecified transition faults. Phase 1 starts from the test set T_2 . It produces test sets that are denoted by $T_3^1, T_4^1, \dots, T_8^1$. These test sets are generated by the test compaction procedure for increasing values of L without considering optimistic unspecified transition faults. In particular, optimistic unspecified transition faults are ignored during the set covering procedure. Fault simulation is carried out for optimistic unspecified transition faults to obtain the fault coverage; however, this fault coverage is not used for guiding the test compaction procedure.

Phase 2 increases the coverage of optimistic unspecified transition faults without attempting to achieve test compaction. Phase 2 starts from the test set T_8^1 obtained at the end of phase 1. It produces test sets that are denoted by $T_3^2, T_4^2, \dots, T_8^2$. These test sets are generated for increasing values of L as follows. As in the test compaction procedure, tests are added to T_L^2 based on T_2 and based on pairs of tests in T_2 . However, every test is simulated using standard and optimistic unspecified transition faults before it is actually added to T_L^2 . A test is added only if it increases the fault coverage. The set covering procedure is applied to T_L^2 to remove tests that are not necessary for achieving the fault coverage. However, without creating a large pool of L -cycle tests in T_L^2 , the ability of the procedure to achieve test compaction is limited.

Table 1. Three-Phase Procedure

circuit	sv	pi	ph	L	tests		f.c.		cycles	ratio	ntime
							str	xtr			
s1423	74	17	0	2	69	2.00	73.963	74.104	5318	1.000	1.00
s1423	74	17	1	8	62	3.89	74.069	78.742	4903	0.922	93.91
s1423	74	17	2	8	68	4.12	74.069	79.866	5386	1.013	192.32
s1423	74	17	3	7	64	4.42	74.069	79.902	5093	0.958	326.86
s13207	669	31	0	2	349	2.00	79.983	80.769	234848	1.000	1.00
s13207	669	31	1	8	313	3.38	79.983	82.772	211124	0.899	96.48
s13207	669	31	2	8	380	3.63	79.983	83.436	256270	1.091	201.79
s13207	669	31	3	8	353	4.69	79.983	83.466	238482	1.015	387.47
b03	30	5	0	2	23	2.00	85.677	85.677	766	1.000	1.00
b03	30	5	1	8	16	5.62	85.677	87.760	600	0.783	97.17
b03	30	5	2	8	17	5.41	85.677	88.151	632	0.825	184.50
b03	30	5	3	6	16	5.12	85.677	88.151	592	0.773	296.25
b07	51	2	0	2	64	2.00	70.300	70.403	3443	1.000	1.00
b07	51	2	1	8	41	5.24	70.868	83.110	2357	0.685	81.22
b07	51	2	2	8	52	5.37	70.868	85.279	2982	0.866	168.94
b07	51	2	3	8	51	5.51	70.971	85.382	2933	0.852	321.79
b08	21	10	0	2	45	2.00	73.223	73.697	1056	1.000	1.00
b08	21	10	1	8	28	5.71	73.223	80.450	769	0.728	99.88
b08	21	10	2	8	37	6.14	73.223	84.597	1025	0.971	188.46
b08	21	10	3	8	33	6.79	73.223	84.834	938	0.888	389.90
b11	30	8	0	2	73	2.00	81.585	81.639	2366	1.000	1.00
b11	30	8	1	8	42	5.74	81.967	88.470	1531	0.647	89.93
b11	30	8	2	8	57	5.60	81.967	92.077	2059	0.870	176.61
b11	30	8	3	8	53	6.06	81.967	92.186	1941	0.820	348.65
b14	247	33	0	2	207	2.00	72.043	72.043	51790	1.000	1.00
b14	247	33	1	8	194	3.32	72.206	76.962	48810	0.942	95.19
b14	247	33	2	8	279	3.83	72.363	80.297	70229	1.356	196.30
b14	247	33	3	8	276	4.24	72.526	80.751	69588	1.344	380.81
sasc	117	15	0	2	47	2.00	85.477	85.542	5710	1.000	1.00
sasc	117	15	1	8	43	4.23	85.477	87.631	5330	0.933	109.22
sasc	117	15	2	8	51	4.24	85.477	88.251	6300	1.103	203.42
sasc	117	15	3	7	48	4.67	85.477	88.251	5957	1.043	366.68
usb_phy	98	14	0	2	52	2.00	88.958	88.958	5298	1.000	1.00
usb_phy	98	14	1	8	48	2.98	88.958	91.191	4945	0.933	132.91
usb_phy	98	14	2	8	61	3.79	88.958	93.424	6307	1.190	225.84
usb_phy	98	14	3	8	58	4.45	88.958	93.755	6040	1.140	467.15

Phase 3 applies the test compaction procedure so as to achieve test compaction, as well as detect additional optimistic unspecified transition faults. Phase 3 starts from the test set T_8^2 obtained at the end of phase 2. It produces test sets denoted by T_3^3 , T_4^3 , ..., T_8^3 . These test sets are generated for increasing values of L without using fault simulation during the construction of the pool. The large pool that results supports test compaction. In addition, the set covering procedure considers standard as well as optimistic unspecified transition faults.

4 EXPERIMENTAL RESULTS

The test compaction procedure is applied to benchmark circuits as described in this section. In all the experiments, the test set T_2 is a compact two-cycle broadside test set for standard transition faults.

4.1 Three-Phase Procedure

The three-phase test compaction procedure is applied first to several benchmark circuits in order to demonstrate the ability of the procedure to reduce the number of clock cycles as well as increase the optimistic unspecified transition fault coverage. In every one of the three phases, the procedure generates test sets T_L for $L = 3, 4, \dots, 8$.

Table 2. Single-Phase Procedure ($\Sigma_L < \Sigma_2$ for $L > 2$)

circuit	sv	pi	ph	L	tests	ave	f.c.		cycles	ratio	ntime
							str	xtr			
s1423	74	17	0	2	69	2.00	73.963	74.104	5318	1.000	1.00
s1423	74	17	3	13	62	8.65	74.069	81.131	5198	0.977	402.36
s1423	74	17	3	16	62	10.05	74.069	81.342	5285	0.994	596.05
s5378	179	35	0	2	180	2.00	77.781	77.781	32759	1.000	1.00
s5378	179	35	3	14	172	3.80	77.781	77.989	31621	0.965	494.78
s35932	1728	35	0	2	30	2.00	71.800	71.800	53628	1.000	1.00
s35932	1728	35	3	13	15	10.00	71.800	73.492	27798	0.518	226.41
s9234	228	19	0	2	355	2.00	76.543	78.276	81878	1.000	1.00
s9234	228	19	3	14	348	4.50	76.543	79.337	81139	0.991	579.58
s9234	228	19	3	16	349	4.79	76.543	79.375	81472	0.995	743.71
s13207	669	31	0	2	349	2.00	79.983	80.769	234848	1.000	1.00
s13207	669	31	3	15	342	7.87	79.983	83.895	232158	0.989	594.74
s13207	669	31	3	16	346	8.24	79.983	83.902	234994	1.001	674.66
b03	30	5	0	2	23	2.00	85.677	85.677	766	1.000	1.00
b03	30	5	3	11	17	5.88	85.677	88.151	640	0.836	317.57
b05	34	2	0	2	104	2.00	76.344	76.445	3778	1.000	1.00
b05	34	2	3	15	61	12.00	76.512	86.962	2840	0.752	453.95
b05	34	2	3	16	63	12.75	76.546	87.063	2979	0.789	507.29
b07	51	2	0	2	64	2.00	70.300	70.403	3443	1.000	1.00
b07	51	2	3	14	41	9.76	71.384	87.087	2542	0.738	399.58
b07	51	2	3	16	41	10.73	71.384	87.138	2582	0.750	509.48
b08	21	10	0	2	45	2.00	73.223	73.697	1056	1.000	1.00
b08	21	10	3	4	38	3.47	73.223	80.332	951	0.901	43.68
b08	21	10	3	14	34	9.85	73.223	85.900	1070	1.013	534.07
b09	28	2	0	2	24	2.00	79.056	79.056	748	1.000	1.00
b09	28	2	3	4	19	3.26	79.056	80.826	622	0.832	48.00
b09	28	2	3	16	28	8.79	79.351	86.431	1058	1.414	723.40
b11	30	8	0	2	73	2.00	81.585	81.639	2366	1.000	1.00
b11	30	8	3	8	51	6.04	81.967	91.694	1868	0.790	162.69
b11	30	8	3	16	47	10.70	81.967	92.350	1943	0.821	536.42
aes_core	530	258	0	2	311	2.00	96.177	96.177	165982	1.000	1.00
aes_core	530	258	3	16	124	13.65	96.177	96.340	67943	0.409	534.29
simple_spi	131	15	0	2	78	2.00	76.702	76.806	10505	1.000	1.00
simple_spi	131	15	3	14	68	7.18	77.173	80.052	9527	0.907	461.50
simple_spi	131	15	3	16	70	7.84	77.330	80.236	9850	0.938	592.54
spi	229	45	0	2	865	2.00	82.587	84.375	200044	1.000	1.00
spi	229	45	3	15	803	6.04	83.314	86.063	188966	0.945	620.82
spi	229	45	3	16	808	6.36	83.322	86.080	190401	0.952	702.79
systemcaes	670	258	0	2	202	2.00	88.735	88.735	136414	1.000	1.00
systemcaes	670	258	3	12	195	9.55	88.735	94.897	133182	0.976	428.22
systemcaes	670	258	3	16	195	12.32	88.735	95.237	133723	0.980	690.23
systemcdes	190	130	0	2	91	2.00	96.089	96.089	17662	1.000	1.00
systemcdes	190	130	3	16	30	14.43	96.089	96.355	6323	0.358	330.25

Table 1 describes the following test sets. The first row for every circuit describes the test set T_2 . The second row describes the test set T_8^1 obtained at the end of phase 1. The third row describes the test set T_8^2 obtained at the end of phase 2. The fourth row describes the final test set T_L^3 of phase 3. The final test set is obtained for $L < 8$ if increasing L further does not increase the fault coverage or reduce the number of clock cycles.

For every test set, after the circuit name, column *sv* shows the number of state variables, and column *pi* shows the number of primary inputs. Column *ph* shows the phase of the procedure, where a zero stands for the initial test set. Column *L* shows the value of L for which T_L is computed.

Column *tests* shows the number of tests in T_L . Column *ave* shows the average number of functional capture cycles in a test of T_L .

Column *f.c.* shows the standard (*str*) and optimistic unspecified (*xtr*) transition fault coverage of T_L .

Table 3. Single-Phase Procedure ($\Sigma_L \geq \Sigma_2$ for $L > 2$)

circuit	sv	pi	ph	L	tests	ave	f.c.		cycles	ratio	ntime
							str	xtr			
s38417	1636	28	0	2	648	2.00	97.106	97.123	1063060	1.000	1.00
s38417	1636	28	3	16	672	4.94	97.154	97.586	1104350	1.039	1367.73
b14	247	33	0	2	207	2.00	72.043	72.043	51790	1.000	1.00
b14	247	33	3	16	261	6.97	72.357	81.106	66532	1.285	666.80
b15	447	36	0	2	488	2.00	81.115	81.149	219559	1.000	1.00
b15	447	36	3	16	627	11.34	83.267	89.535	287824	1.311	1027.68
b20	494	33	0	2	306	2.00	79.651	79.651	152270	1.000	1.00
b20	494	33	3	16	432	7.82	80.519	89.136	217282	1.427	686.94
sasc	117	15	0	2	47	2.00	85.477	85.542	5710	1.000	1.00
sasc	117	15	3	16	49	10.43	85.477	88.544	6361	1.114	729.10
tv80	359	13	0	2	658	2.00	82.023	82.034	237897	1.000	1.00
tv80	359	13	3	16	950	7.76	83.025	93.509	348783	1.466	751.14
usb_phy	98	14	0	2	52	2.00	88.958	88.958	5298	1.000	1.00
usb_phy	98	14	3	15	56	7.59	88.958	94.003	6011	1.135	745.11
wb_dma	523	215	0	2	175	2.00	75.042	75.048	92398	1.000	1.00
wb_dma	523	215	3	16	192	4.14	75.381	76.093	101733	1.101	701.76

Column *cycles* shows the number of clock cycles required for applying T_L , Σ_L . Column *ratio* shows the ratio Σ_L/Σ_2 .

Column *ntime* shows the normalized run time for the test compaction procedure. To obtain the normalized run time, the total run time for all the values of L , up to the one reported, is divided by the run time for fault simulation with fault dropping of F under T_2 .

The following points can be seen from Table 1. The use of multicycle broadside tests sometimes increases the coverage of standard transition faults. However, the increase is typically small, and it is not likely to be sufficient for guiding the generation of multicycle tests that improve the defect coverage.

Even without targeting optimistic unspecified transition faults, phase 1 of the test compaction procedure increases their coverage as it increases the number of functional capture cycles. However, a further, significant increase is typically obtained in phase 2, when the faults are targeted directly.

Even when the coverage of unspecified transition faults increases significantly, the use of multicycle tests in phase 3 typically reduces the number of clock cycles required for applying the test set. In many cases, this results in $\Sigma_L < \Sigma_2$ for $L > 2$.

4.2 Single-Phase Procedure

The intended use of the test compaction procedure is as a single-phase procedure where only phase 3 is applied to achieve test compaction and increase the optimistic unspecified transition fault coverage simultaneously.

The procedure is applied to generate test sets T_L for $L = 3, 4, \dots, 16$. With a higher bound on L , the procedure has more flexibility to increase the unspecified transition fault coverage while achieving test compaction.

The results are shown in Tables 2 and 3 in the same format as Table 1. The test sets included in Tables 2 and 3 are the following. The first row for every circuit describes the test set T_2 . In Table 2, the second row describes the test set T_L with the smallest number of clock cycles, Σ_L , such that $\Sigma_L < \Sigma_2$. A test set with $\Sigma_L < \Sigma_2$ is not obtained for the circuits in Table 3 because the increased fault coverage requires an increased number of clock cycles, and the increase in the number of clock cycles is accepted. The last row describes the final test set T_L , with the largest coverage of optimistic unspecified transition faults, and the smallest number of clock cycles. In some cases, a test set with $\Sigma_L < \Sigma_2$ is also the one with the smallest number of clock cycles and the largest

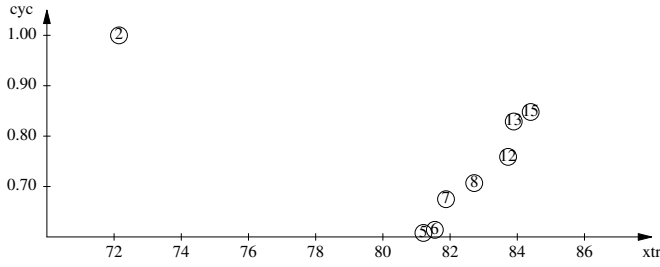


Fig. 3. Single-phase procedure

coverage of optimistic unspecified transition faults. In this case, only one test set, other than the initial test set, is included in Table 2.

From Tables 2 and 3 it can be seen that the single-phase procedure increases the coverage of optimistic unspecified transition faults significantly. With a larger value of L , the fault coverage is higher for many circuits in Tables 2 and 3 than in Table 1, and the number of clock cycles is lower.

The ability of the procedure to achieve test compaction can be seen from the fact that, in many cases, $\Sigma_L < \Sigma_2$ is obtained even with an increased fault coverage.

A value of L can be selected for a circuit based on the number of clock cycles and the fault coverage. Tables 2 and 3 show at most two test sets that are produced by the test compaction procedure. Additional, intermediate options are available, with intermediate numbers of clock cycles and fault coverage.

To illustrate this point, Figure 3 shows the results of the test compaction procedure when it is applied to benchmark circuit *s298* to produce test sets $T_3, T_4, \dots, T_{16}$. The horizontal axis shows the coverage of optimistic unspecified transition faults. The vertical axis shows the reduction in the number of clock cycles required for applying the test set, Σ_L/Σ_2 . The value of L is shown inside the circle for the test set T_L , where $2 \leq L \leq 16$. A value of L is omitted from Figure 3 if the procedure obtains a test set with a higher optimistic unspecified transition fault coverage and a lower number of clock cycles for a higher value of L . In this case, the higher value of L is preferred, and the lower value is omitted.

Figure 3 demonstrates that the procedure produces several test sets, representing different points in the tradeoff between the optimistic unspecified transition fault coverage and the number of clock cycles.

5 CONCLUDING REMARKS

This paper suggested that the generation of multicyle tests for test compaction should incorporate a fault model, whose fault coverage increases when multicyle tests are used, in order to materialize the benefit of multicyle tests in increasing the defect coverage. To ensure that the computational complexity of test compaction for transition faults is not increased by a fault model with a large number of faults, or faults with complex behaviors, optimistic unspecified transition faults were added to the transition fault model to achieve this goal. The paper described a test compaction procedure that produces multicyle tests for transition faults of both types. Experimental results for benchmark circuits demonstrated the improvement in both parameters.

REFERENCES

- [1] Z. Barzilai and B. Rosen, "Comparison of AC Self-Testing Procedures", in Proc. Intl. Test Conf., 1983, pp. 89-94.
- [2] G. L. Smith, "Model for Delay Faults Based Upon Paths", in Proc. Intl. Test Conf., 1985, pp. 342-349.

- [3] J. L. Carter, V. S. Iyengar and B. K. Rosen, "Efficient Test Coverage Determination for Delay Faults", in Proc. Intl. Test Conf., 1987, pp. 418-427.
- [4] K.-T. Cheng, "Transition Fault Testing for Sequential Circuits", IEEE Trans. on Computer-Aided Design, Dec. 1993, pp. 1971-1983.
- [5] I. Pomeranz and S. M. Reddy, "Unspecified Transition Faults: A Transition Fault Model for At-Speed Fault Simulation and Test Generation", IEEE Trans. on Computer-Aided Design, Jan. 2008, pp. 137-146.
- [6] A. Sreedhar, A. Sanyal and S. Kundu, "On Modeling and Testing of Lithography Related Open Faults in Nano-CMOS Circuits", in Proc. Design, Autom. and Test in Europe Conf., 2008, pp. 616-621.
- [7] M. O. Simsir, A. Bhoj and N. K. Jha, "Fault Modeling for FinFET Circuits", in Proc. Intl. Symp. on Nanoscale Architectures, 2010, pp. 41-46.
- [8] J. Zha, X. Cui and C. L. Lee, "Modeling and Testing of Interference Faults in the Nano NAND Flash Memory", in Proc. Design, Automation & Test in Europe Conf., 2012, pp. 527-531.
- [9] D. Xiang, Z. Chen and L.-T. Wang, "Scan Flip-Flop Grouping to Compress Test Data and Compact Test Responses for Launch-on-Capture Delay Testing", ACM Trans. on Design Automation, Vol. 17, No. 2, April 2012, Article No. 18.
- [10] S.-Y. Huang, Y.-H. Lin, K.-H. Tsai, W.-T. Cheng, S. Sunter, Y.-F. Chou and D.-M. Kwai, "Small Delay Testing for TSVs in 3-D ICs", in Proc. Design Automation Conf., 2012, pp. 1031-1036.
- [11] A. Mondal, P. P. Chakrabarti and P. Dasgupta, "Symbolic-Event-Propagation-Based Minimal Test Set Generation for Robust Path Delay Faults", ACM Trans. on Design Automation, Vol. 17 No. 4, Oct. 2012, Article No. 47.
- [12] M. Sauer, A. Czuto, I. Polian and B. Becker, "Small-delay-fault ATPG with Waveform Accuracy", in Proc. Intl. Conf. on Computer-Aided Design, 2012, pp. 30-36.
- [13] W. Zhao, J. Ma, M. Tehranipoor and S. Chakravarty, "Power-safe Application of TDF Patterns to Flip-chip Designs during Wafer Test", ACM Trans. on Design Automation, Vol. 18, No. 3, July 2013, Article No. 43.
- [14] S. Di Carlo, G. Gambardella, P. Prinetto, D. Rolfo and P. Trotta, "SATTA: A Self-Adaptive Temperature-Based TDF Awareness Methodology for Dynamically Reconfigurable FPGAs", ACM Trans. on Reconfigurable Technology and Systems, Vol. 8, No. 1, Feb. 2015, Article No. 1.
- [15] Y. Zhang, Z. Peng, J. Jiang, H. Li and M. Fujita, "Temperature-aware Software-based Self-testing for Delay Faults", in Proc. Design, Automation & Test in Europe Conf., 2015, pp. 423-428.
- [16] H. G. Mohammadi, P.-E. Gaillardon and G. De Micheli, "Fault Modeling in Controllable Polarity Silicon Nanowire Circuits", in Proc. Design, Automation & Test in Europe Conf., 2015, pp. 453-458.
- [17] A. S. Trinadh, S. Potluri, S. B. Ch., V. Kamakoti and S. G. Singh, "Optimal Don't Care Filling for Minimizing Peak Toggles During At-Speed Stuck-At Testing", ACM Trans. on Design Automation, Vol. 23, No. 1, Oct. 2017, Article No. 5.
- [18] P. C. Maxwell, R. C. Aitken, K. R. Kollitz and A. C. Brown, "IDDQ and AC Scan: The War Against Unmodelled Defects", in Proc. Intl. Test Conf., 1996, pp. 250-258.
- [19] J. Rearick, "Too Much Delay Fault Coverage is a Bad Thing", in Proc. Intl. Test Conf., 2001, pp. 624-633.
- [20] X. Lin and R. Thompson, "Test Generation for Designs with Multiple Clocks", in Proc. Design Autom. Conf., 2003, pp. 662-667.
- [21] G. Bhargava, D. Meehl and J. Sage, "Achieving Serendipitous N-Detect Mark-Offs in Multi-Capture-Clock Scan Patterns", in Proc. Intl. Test Conf, 2007, Paper 30.2.
- [22] I. Park and E. J. McCluskey, "Launch-on-Shift-Capture Transition Tests", in Proc. Intl. Test Conf., 2008, pp. 1-9.
- [23] E. K. Moghaddam, J. Rajski, S. M. Reddy and M. Kassab, "At-Speed Scan Test with Low Switching Activity", in Proc. VLSI Test Symp., 2010, pp. 177-182.
- [24] I. Pomeranz, "Static Test Compaction for Mixed Broadside and Skewed-Load Transition Fault Test Sets", IET Computers & Digital Techniques, Jan. 2013, pp. 21-28.
- [25] D. Erb, K. Scheibler, M. Sauer, S. M. Reddy and B. Becker, "Multi-cycle Circuit Parameter Independent ATPG for Interconnect Open Defects", in Proc. VLSI Test Symp., 2015, pp. 1-6.
- [26] I. Pomeranz, "A Multi-Cycle Test Set Based on a Two-Cycle Test Set with Constant Primary Input Vectors", IEEE Trans. on Computer-Aided Design, July 2015, pp. 1124-1132.
- [27] S. Y. Lee and K. K. Saluja, "Test Application Time Reduction for Sequential Circuits with Scan", IEEE Trans. on Computer-Aided Design, Sept. 1995, pp. 1128-1140.
- [28] I. Pomeranz and S. M. Reddy, "Static Test Compaction for Scan-Based Designs to Reduce Test Application Time", in Proc. Asian Test Symp., 1998, pp. 198-203.

Received January 2019