Single Test Type to Replace Broadside and Skewed-Load Tests for Transition Faults

Irith Pomeranz, *Fellow, IEEE*, and Xijiang Lin, *Member, IEEE*

***Abstract*— The use of both broadside (launch-on-capture) and skewed-load (launch-on-shift) tests for delay faults results in increased delay fault coverage and better test compaction than the use of a single test type. Two-cycle broadside and skewed-load tests differ in the sequence of length two applied to the scan-enable input between the scan-in and scan-out operations of a test. Considering a circuit with a single clock domain, the question that this article attempts to answer is whether it is possible to generate a complete test set for transition faults where all the tests use the same scan-enable sequence of length three or more. The use of a single scan-enable sequence simplifies the test application process. Experimental results demonstrate that there is a significant number of benchmark circuits for which a test set with a single scan-enable sequence achieves the same transition fault coverage as a test set that consists of both broadside and skewed-load tests. For other benchmark circuits, a small loss in transition fault coverage compared with the use of both test types allows a single scan-enable sequence to be used.**

***Index Terms*— Broadside tests, full-scan circuits, multicycle tests, skewed-load tests, test generation, transition faults.**

I. Introduction

DELAY faults require two-cycle tests to create $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions that are needed for activating the faults. Two types of two-cycle tests exist for standard scan circuits, broadside (launch-on-capture), and skewed-load (launch-on-shift) tests [1], [2]. The tests differ in the types of clock cycles between the scan-in and scan-out operations of a test. Under a broadside test, both clock cycles are functional capture cycles. Under a skewed-load test, the first clock cycle is a scan shift cycle and the second is a functional capture cycle. Suppose that a 0 on the scan-enable input corresponds to a functional capture cycle, and a 1 corresponds to a scan shift cycle. Let SE_i be the sequence assigned to the scan-enable input between the scan operations of a test t_i. With this notation, a broadside test has a scan-enable sequence $SE_i = 00$, and a skewed-load test has $SE_i = 10$.

With both test types, the first clock cycle after the scan-in operation is applied under a slow clock, and the second clock cycle is applied under a fast clock. Consequently, a skewed-load test requires the scan-enable input to change from 1 to 0 at the circuit speed. This requirement can be addressed as described in [3] and [4].

Manuscript received August 28, 2020; revised October 28, 2020; accepted November 9, 2020. Date of publication December 1, 2020; date of current version January 28, 2021. *(Corresponding author: Irith Pomeranz.)*

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

Xijiang Lin is with Mentor, A Siemens Business, Wilsonville, OR 97070 USA (e-mail: xijiang_lin@mentor.com).

Digital Object Identifier 10.1109/TVLSI.2020.3038368

The motivation for allowing both types of tests to be included in a test set is discussed next. Only circuits with single clock domains are considered in this article.

It was shown in [1] and [2] that a delay fault may only be detectable by a broadside or skewed-load test. Consequently, the delay fault coverage achievable by a test set that consists of both test types is higher than the delay fault coverage achievable by a test set that consists of only broadside or only skewed-load tests. In addition, the achievable level of test compaction is higher when both test types are available [5]. In general, test sets that combine different test types and fault models are needed for addressing the variety of defects that can occur [6]–[12].

Considering the level of test compaction achieved by using both broadside and skewed-load tests, even higher levels of test compaction are achievable if the scan-enable input is allowed to assume arbitrary sequences as discussed next.

In the approach referred to as transparent-scan in [13], a test set is replaced by a sequence of scan shift and functional capture cycles that is not necessarily partitioned into tests. This allows scan operations of reduced lengths, or combinations of scan shift and functional capture cycles, to be used for bringing the circuit into a state where a fault can be detected, and observing fault effects. The sequence consists of a significantly fewer clock cycles than a test set that can be partitioned into tests.

In [14], skewed-load and broadside tests are combined into three-cycle tests, with three cycles between scan-in and scan-out operations. The corresponding scan-enable sequence is a combination of 10 and 00, e.g., $SE_i = 100$. A test can detect more faults than a broadside or skewed-load test alone, contributing to test compaction.

In [15], transparent-scan is restricted to the clock cycles between the scan-in and scan-out operations of a test. Instead of using tests with $SE_i = 00$ or $SE_i = 10$ as in [1] and [2], or tests with $SE_i = 100$ as in [14], bounded transparent-scan in [15] allows any sequence to be used for the scan-enable input between the scan-in and scan-out operations of a test. For example, a five-cycle test t_0 may have $SE_0 = 10110$, implying that after the scan-in operation, there is a scan shift cycle, a functional capture cycle, two additional scan shift cycles, and one additional functional capture cycle before the scan-out operation. In the same test set, a different test t_1 may have $SE_1 = 011100$. The use of multicycle tests contributes to test compaction since a test can detect more faults than a two-cycle test [16]–[19].

The question that this article attempts to answer is whether it is possible to generate a complete test set for delay faults

(transition faults in this article) where all the tests use the same scan-enable sequence SE_j. The use of a single scan-enable sequence for all the tests in a test set simplifies the test application process that does not need to accommodate different test types with different scan-enable sequences. This preserves the common practice of using a single test type (either broadside or skewed-load) while allowing the fault coverage of two test types to be achieved. A higher fault coverage than possible with a single test type is important for higher coverage of delay defects. The single scan-enable sequence can be stored on-chip for at-speed test application, or implementation as described in [3] and [4] can be used to ensure that it can be applied at speed. With state-of-the-art synthesis flows, where the synthesis process is iterative, the logic related to the scan-enable sequence can be synthesized after test generation determines which scan-enable sequence to use. This ensures that the selected scan-enable sequence can be accommodated similar to the case where skewed-load tests are used. Compared with a solution that allows multiple scan-enable sequences to be used, which may be the case with $LBIST$, the use of a single scan-enable sequence (or a reduced number of scan-enable sequences) reduces the required hardware support.

This article studies this question through two experiments. Both experiments use a comprehensive set $SE = \{SE_0, SE_1, \ldots, SE_{m-1}\}$ of possible scan-enable sequences. Test generation is performed with every sequence in SE separately. The goal is to find the shortest sequence that is sufficient on its own for detecting transition faults. In the best case, the sequence will result in the detection of all the faults that are detectable by broadside and skewed-load tests. A small loss in fault coverage may also be acceptable to benefit from the simplicity of a single scan-enable sequence. It is important to note in this regard that the common practice is to use only broadside or only skewed-load tests. In this case, the loss of fault coverage compared with the use of both test types is larger.

The first experiment carried out in this article is based on exhaustive test sets for small finite-state machine benchmarks. This experiment provides complete and accurate answers regarding effective scan-enable sequences. The second experiment is based on test generation and it is applied to larger benchmark circuits. It uses incremental test generation based on the following experimental observation.

Suppose that test generation is carried out for scan-enable sequences $SE_0, SE_1, \ldots$ of increasing length. Let the test set computed for SE_j be $T(SE_j)$. This implies that all the tests in $T(SE_j)$ use SE_j as their scan-enable sequence. Suppose that $T(SE_j)$ achieves a higher fault coverage than any one of $T(SE_0), \ldots, T(SE_{j-1})$. In this case, it provides an effective starting point for the generation of $T(SE_{j+1})$, $T(SE_{j+2}), \ldots$ A starting point for $T(SE_k)$, where $k > j$, is obtained by updating the tests in $T(SE_j)$ to use SE_k as their scan-enable sequence, while keeping scan-in states and primary input vectors as discussed later. This sharing of test data between different scan-enable sequences reduces the overall test generation effort, and allows a large number of scan-enable sequences to be considered.

Experimental results also indicate that scan-enable sequences selected for different circuits have certain properties that are common across the circuits. These properties are used for reducing the set SE of possible scan-enable sequences, thus reducing the test generation effort required for finding an effective scan-enable sequence.

It should be noted that the fault simulation procedure from [15] allows a test with an arbitrary scan-enable sequence to be simulated. Commercial tools also support fault simulation and the generation of tests with arbitrary scan-enable sequences. In this article, it is assumed that output values (values of primary outputs and scan chain outputs) are observed at-speed between the scan operations of a test, starting from the second clock cycle. It is possible to use output compaction logic to capture output values at-speed [20]. Although the article focuses on transition faults, it is possible to consider other fault models such as path-delay faults, cell-aware, or gate-exhaustive faults that model delay defects. By using multicycle tests, the coverage of delay defects is likely to increase.

Several extensions of the approach described in this article are possible but are not considered here: 1) extending the approach in this article to a circuit with multiple clock domains implies that a single control sequence would be used for a test set. Such a control sequence would allow each clock domain to be tested separately in its turn under a single scan-enable sequence; 2) test points may be used to address the small loss of fault coverage that sometimes occurs when a single scan-enable sequence is used to replace broadside and skewed-load tests; and 3) the ability to detect faults is necessary for supporting fault diagnosis. Moreover, achieving the fault coverage of a broadside and skewed-load test set using a single scan-enable sequence is likely to provide the same diagnosis quality as a test set that consists of both test types. Results of diagnostic fault simulation are included to support this point.

This article is organized as follows. The advantages of using both broadside and skewed-load tests are demonstrated in Section II. The extended set of scan-enable sequences considered for both studies and the format of a test are described in Section III. Section IV discusses the study based on exhaustive test sets and provides experimental results. Section V discusses the study based on test generation and provides experimental results. Section VI discusses the common properties of scan-enable sequences for different circuits and presents experimental results using a reduced set. The target faults for all the experiments are transition faults.

II. Broadside and Skewed-Load Tests

To demonstrate the advantages of a test set that consists of both broadside and skewed-load tests, the study of exhaustive test sets was first applied to compute three test sets: a test set $T_{\text{brd+skw}}$ that consists of both broadside and skewed-load tests, a test set T_{brd} that consists only of broadside tests, and a test set T_{skw} that consists only of skewed-load tests. The computation of exhaustive test sets is described in Section IV. The test sets are complete in that they detect every fault that can be detected by the allowed test type(s). Forward-looking

TABLE I
BROADSIDE AND SKEWED-LOAD TEST SETS

			brd+skw		brd			skw		
circuit	sv	pi	f.c.	tests	f.c.	diff	tests	f.c.	diff	tests
dk27	3	3	83.607	13	79.508	4.098	11	77.049	6.557	12
ex4	4	7	89.222	30	82.335	6.886	20	81.138	8.084	27
fetch	5	11	88.319	51	84.758	3.561	44	80.769	7.550	42
log	5	11	89.109	47	83.333	5.776	39	82.508	6.601	40
rie	5	11	88.402	75	83.763	4.639	57	83.162	5.241	68
dvram	6	10	93.567	65	90.632	2.935	56	85.102	8.465	55
nucpwr	5	15	89.655	70	85.022	4.634	51	80.280	9.375	54
dk512	4	3	86.087	23	71.304	14.783	14	80.435	5.652	21
ex2	5	4	82.581	54	71.452	11.129	41	76.774	5.806	50
ex3	4	4	80.333	27	74.667	5.667	19	76.000	4.333	24
mark1	4	6	79.630	23	72.751	6.878	20	73.280	6.349	19
modulo12	4	3	83.607	13	77.049	6.557	11	80.328	3.279	14
opus	4	7	82.143	30	76.099	6.044	30	76.648	5.495	32
shiftreg	3	3	84.615	9	73.077	11.538	7	78.846	5.769	7

reverse order fault simulation is applied to remove unnecessary tests from the test set.

Information about the test sets for finite-state machine benchmarks is shown in Table I. The circuits are such that all three test sets achieve a fault coverage of at least 70%. The constraints of the test type(s) prevent all the faults from being detected. For the circuits in the first part of Table I the fault coverage of T_{brd} is higher than that of T_{skw}. For the circuits in the second part of Table I the fault coverage of T_{skw} is higher than that of T_{brd}.

After the circuit name, column sv of Table I shows the number of state variables, and column pi shows the number of primary inputs. Column $brd{+}skw$ shows information about the test set $T_{\text{brd+skw}}$. Column brd shows information about the test set T_{brd}. Column skw shows information about the test set T_{skw}. In every case, subcolumn $f.c.$ shows the transition fault coverage. For T_{brd} and T_{skw}, subcolumn $diff$ shows the difference between the transition fault coverage and that of $T_{\text{brd+skw}}$. Subcolumn $tests$ shows the number of tests.

Table I demonstrates that the transition fault coverage of $T_{\text{brd+skw}}$ can be significantly higher than that of T_{brd} or T_{skw}. Between T_{brd} and T_{skw}, there is no clear choice for one test set whose fault coverage is typically higher. Overall, Table I supports the use of both test types to achieve the highest possible transition fault coverage.

III. SCAN-ENABLE SEQUENCES

This section discusses the scan-enable sequences considered in this article, and the format of a test.

A broadside test t_i is defined by a scan-in state s_i, a primary input vector v_i, and a scan-enable sequence $\text{SE}_i = 00$. The same primary input vector v_i is used for both clock cycles of the test. This is a common choice for addressing tester limitations in changing the primary input vector at-speed during a test. This choice is used for all the tests considered in this article.

A skewed-load test t_i requires, in addition, a scan-in vector for the scan shift cycle that follows the scan-in operation. In general, the scan-in sequence for a test t_i is denoted by SI_i. If a test t_i has $\lambda_i \geq 2$ clock cycles between its scan-in and scan-out operations, we have that $\text{SE}_i = \text{SE}_i(0)\text{SE}_i(1)\ldots\text{SE}_i(\lambda_i - 1)$ and $\text{SI}_i = \text{SI}_i(0)\text{SI}_i(1)\ldots\text{SI}_i(\lambda_i - 1)$. For $0 \leq u < \lambda_i$, if $\text{SE}_i(u) = 0$, the clock cycle is a functional capture cycle, and the scan-in vector does not affect fault detection. For simplicity of discussion, suppose that the circuit has a single scan chain. By default, $\text{SE}_i(u) = 0$ is associated with $\text{SI}_i(u) = 0$. With $\text{SE}_i(u) = 1$, the clock cycle is a scan shift cycle. Either $\text{SI}_i(u) = 0$ or $\text{SI}_i(u) = 1$ can be assigned as the scan-in value during the scan shift cycle.

Using this notation, a test is represented as $t_i = \langle s_i, v_i, \text{SE}_i, \text{SI}_i\rangle$. A broadside test is represented as $t_i = \langle s_i, v_i, 00, 00\rangle$, and a skewed-load test as $t_i = \langle s_i, v_i, 10, \text{SI}_i(0)0\rangle$.

With two-cycle tests, it is necessary to use both broadside and skewed-load tests, or both $\text{SE}_i = 00$ and $\text{SE}_i = 10$, to achieve the maximum transition fault coverage achievable in the standard scan circuit. This article considers the use of a single scan-enable sequence SE_j of length $\lambda_j > 2$ with which it is possible to achieve the maximum transition fault coverage. In the resulting test set $T(\text{SE}_j)$, every test t_i uses SE_j as its scan-enable sequence, i.e., $\text{SE}_i = \text{SE}_j$ for every test t_i.

There are 2^{λ_j} scan-enable sequences of length λ_j. To achieve the fault coverage of broadside and skewed-load tests, it is important to include in the scan-enable sequence both functional capture and scan shift cycles. Additional constraints will be discussed later based on the experimental results obtained for benchmark circuits.

Excluding only the all-0 and all-1 sequences, for $\lambda_j = 3$, the possible scan-enable sequences are 001, 010, 011, 100, 101, and 110. For $\lambda_j = 4$, the possible scan-enable sequences are 0001, 0010, 0011, ..., 1110.

The procedures described in the next sections consider $3 \leq \lambda_j \leq 10$. The upper bound of $\lambda_j = 10$ was selected based on experimental results showing that $\lambda_j < 10$ is typically selected for benchmark circuits.

The extended set of all the possible scan-enable sequences is denoted by $\text{SE} = \{\text{SE}_0, \text{SE}_1, \ldots, \text{SE}_{m-1}\}$, and it includes $2^{\lambda_j} - 2$ sequences for every value of $\lambda_j \geq 3$. For completeness, $\text{SE}_0 = 00, 10$ corresponds to the use of broadside and skewed-load tests. The sequences are arranged by increasing length, and this is also the order by which they are considered for test generation. The procedures described in the next sections

search for the first sequence in SE where the fault coverage of $T(\mathrm{SE}_j)$, for $j > 0$, is the same as the fault coverage of $T(\mathrm{SE}_0)$.

IV. Exhaustive Test Sets

This section describes the use of exhaustive test sets to select a scan-enable sequence SE_j for a circuit.

A. Procedure

The procedure considers the scan-enable sequences from SE in the order by which they appear in SE. Considering $\mathrm{SE}_0 = 00, 10$, an exhaustive broadside and skewed-load test set is computed. The test set, denoted by $T(\mathrm{SE}_0)$, consists of every test of the form $t_i = \langle s_i, v_i, \mathrm{SE}_i, \mathrm{SI}_i \rangle$, where $\mathrm{SE}_i = 00$ or 10. With $\mathrm{SE}_i = 00$, the only option for SI_i is $\mathrm{SI}_i = 00$. With $\mathrm{SE}_i = 10$, there are two options, $\mathrm{SI}_i = 00$ or 10. Thus, there are three options for SE_i and SI_i. Considering a circuit with k state variables, and n primary inputs, there are 2^k options for a scan-in state, and 2^n options for a primary input vector. The number of tests in $T(\mathrm{SE}_0)$ is equal to $3 \cdot 2^{(k+n)}$.

Considering SE_j for $j > 0$, $T(\mathrm{SE}_j)$ consists of every test of the form $t_i = \langle s_i, v_i, \mathrm{SE}_j, \mathrm{SI}_i \rangle$. In this case, there is only one option for SE_i, $\mathrm{SE}_i = \mathrm{SE}_j$. Let u_j be the number of 1s in SE_j. The number of options for SI_i is 2^{u_j}. For example, with $\mathrm{SE}_j = 1100$, the options for SI_i are 0000, 0100, 1000, and 1100. Considering a circuit with k state variables, and n primary inputs, the number of tests in $T(\mathrm{SE}_j)$ is equal to $2^{(k+n+u_j)}$.

After computing the exhaustive test set $T(\mathrm{SE}_j)$, for $0 \leq j < m$, fault simulation followed by forward-looking reverse order fault simulation is applied to remove unnecessary tests from $T(\mathrm{SE}_j)$. The procedure stops with the first scan-enable sequence $\mathrm{SE}_j \in SE$ where $T(\mathrm{SE}_j)$ achieves the fault coverage of $T(\mathrm{SE}_0)$, and $j > 0$.

B. Experimental Results

The results of exhaustive test generation are shown in Tables II and III. The circuits are finite-state machine benchmarks with numbers of state variables and primary inputs that allow exhaustive test sets to be derived for $3 \leq \lambda_j \leq 10$. Table II contains circuits for which $T(\mathrm{SE}_j)$ with $\lambda_j = 3$ achieves the fault coverage of the broadside and skewed-load test set $T(\mathrm{SE}_0)$. Table III contains circuits that require longer scan-enable sequences. The circuits in Table III are arranged by increasing length of the scan-enable sequence selected for them. For the same length, the circuits are arranged in chronological order based on the scan-enable sequence.

For every circuit, the first row corresponds to SE_0. The second row corresponds to the selected scan-enable sequence SE_j.

For every test set, column *sv* shows the number of state variables, and column *pi* shows the number of primary inputs. Column *len* shows the length λ_j of SE_j. For $j > 0$, column *seq* shows the sequence SE_j. Column *f.c.* shows the transition fault coverage of $T(\mathrm{SE}_j)$. Column *tests* shows the number of tests in $T(\mathrm{SE}_j)$. Column *cycles* shows the number of clock cycles required for applying it, including both functional capture and scan shift cycles.

TABLE II
Exhaustive Test Sets, $\lambda_j = 3$

circuit	sv	pi	len	seq	f.c.	tests	cycles
lion	2	4	2		64.706	6	26
lion	2	4	3	010	64.706	6	32
mc	2	5	2		81.538	13	54
mc	2	5	3	010	81.538	9	47
train4	2	4	2		69.118	4	18
train4	2	4	3	010	69.118	4	22
bbsse	4	9	2		74.558	39	238
bbsse	4	9	3	100	74.558	30	214
dk17	3	4	2		78.629	23	118
dk17	3	4	3	100	78.629	17	105
dk27	3	3	2		83.607	13	68
dk27	3	3	3	100	83.607	9	57
dk512	4	3	2		86.087	23	142
dk512	4	3	3	100	86.087	15	109
dvram	6	10	2		93.567	65	526
dvram	6	10	3	100	93.567	46	420
ex3	4	4	2		80.333	27	166
ex3	4	4	3	100	80.333	18	130
ex4	4	7	2		89.222	30	184
ex4	4	7	3	100	89.222	21	151
ex5	3	4	2		81.786	27	138
ex5	3	4	3	100	81.786	18	111
ex6	3	7	2		79.204	36	183
ex6	3	7	3	100	79.204	27	165
ex7	4	4	2		76.351	29	178
ex7	4	4	3	100	76.351	22	158
fetch	5	11	2		88.319	51	362
fetch	5	11	3	100	88.319	35	285
log	5	11	2		89.109	47	334
log	5	11	3	100	89.109	36	293
mark1	4	6	2		79.630	23	142
mark1	4	6	3	100	79.630	17	123
modulo12	4	3	2		83.607	13	82
modulo12	4	3	3	100	83.607	10	74
nucpwr	5	15	2		89.655	70	495
nucpwr	5	15	3	100	89.655	50	405
opus	4	7	2		82.143	30	184
opus	4	7	3	100	82.143	27	193
rie	5	11	2		88.402	75	530
rie	5	11	3	100	88.402	52	421
s8	3	5	2		66.667	14	73
s8	3	5	3	100	66.667	12	75
shiftreg	3	3	2		84.615	9	48
shiftreg	3	3	3	100	84.615	5	33
sse	4	9	2		74.558	39	238
sse	4	9	3	100	74.558	30	214
tav	2	6	2		52.632	11	46
tav	2	6	3	100	52.632	6	32
beecount	3	5	2		62.871	15	78
beecount	3	5	3	101	62.871	15	93
lion9	3	4	2		71.818	12	63
lion9	3	4	3	101	71.818	12	75

Tables II and III demonstrate that, in general, different circuits require different scan-enable sequences to achieve the fault coverage of $T(\mathrm{SE}_0)$. For many of the circuits, the scan-enable sequence is longer than three.

Nevertheless, there are cases where the same scan-enable sequence is effective for different circuits. The most notable example is that of the sequence 100. In addition, the sequence 010 is selected for *lion*, *mc*, and *train*4; the sequence 00110 is selected for *bbtas* and *train*11; and the sequence 0011110 is selected for *firstex* and *keyb*. Common features of the selected sequences are discussed in Section VI.

The number of tests is reduced when SE_j is used. The number of clock cycles increases because each test has more

TABLE III
EXHAUSTIVE TEST SETS, $\lambda_j > 3$

circuit	sv	pi	len	seq	f.c.	tests	cycles
bbtas	3	4	2		75.893	14	73
bbtas	3	4	5	00110	75.893	9	75
train11	4	4	2		71.978	21	130
train11	4	4	5	00110	71.978	16	148
dk14	3	5	2		65.438	26	133
dk14	3	5	5	10100	65.438	18	147
dk15	2	5	2		65.411	23	94
dk15	2	5	5	10100	65.411	13	93
bbara	4	6	2		60.370	24	148
bbara	4	6	6	100110	60.370	20	204
dk16	5	4	2		81.374	88	621
dk16	5	4	6	101000	81.374	42	467
cse	4	9	2		77.628	70	424
cse	4	9	6	101100	77.628	43	434
firstex	4	4	2		70.290	20	124
firstex	4	4	7	0011110	70.290	18	202
keyb	5	9	2		72.041	84	593
keyb	5	9	7	0011110	72.041	69	833
s1a	5	10	2		72.515	98	691
s1a	5	10	7	1001110	72.515	80	965
ex2	5	4	2		82.581	54	383
ex2	5	4	8	00111010	82.581	36	473
donfile	5	4	2		77.208	61	432
donfile	5	4	8	00111110	77.208	61	798

clock cycles between its scan operations. For larger circuits, considered later, scan shift cycles for scan operations dominate the number of clock cycles. In this case, a reduced number of tests leads to a reduced number of clock cycles. Thus, the use of tests with more than two clock cycles leads to test compaction in larger circuits [16]–[19].

V. TEST GENERATION

This section describes the use of test generation to select a scan-enable sequence SE_j for a circuit and generate a test set $T(\mathrm{SE}_j)$.

A. Procedure Overview

An overview of the test generation procedure is given in Fig. 1. Considering $\mathrm{SE}_0 = 00, 10$, a broadside and skewed-load test set $T(\mathrm{SE}_0)$ is assumed to be given. This test set can be generated by any test generation procedure for transition faults that is able to produce both test types.

For scan-enable sequences of length $\lambda_j \geq 3$, sequential test generation is needed to generate tests with three or more clock cycles. To avoid the computational effort of deterministic sequential test generation, the procedure used in this article is simulation-based. This implies that the procedure may not detect all the detectable faults for a given scan-enable sequence SE_j. A test set $T(\mathrm{SE}_j)$, for $j > 2$, is generated in three phases as described next. To demonstrate the possibility of using other test generation and test compaction procedures, results using a commercial tool are presented in Section VI. The main requirement from a test generation or test compaction procedure is that it would handle the scan-enable sequence explicitly.

To describe the three phases of the procedure, and accommodate the fact that phase 3 is iterative, we add an index k to

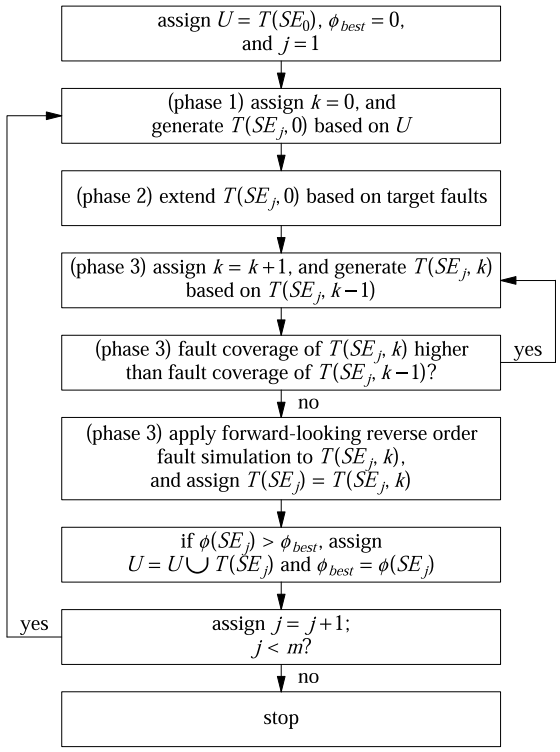

Fig. 1. Overall test generation process.

the test set $T(\mathrm{SE}_j)$. We consider $T(\mathrm{SE}_j, 0)$ in phases 1 and 2, and $T(\mathrm{SE}_j, 1)$ $T(\mathrm{SE}_j, 2)$, ... in phase 3. The final test set is assigned to $T(\mathrm{SE}_j)$.

Phase 1 creates an initial test set $T(\mathrm{SE}_j, 0)$ for SE_j based on tests that were effective for $\mathrm{SE}_0, \ldots, \mathrm{SE}_{j-1}$. This phase is important even if a deterministic test generation procedure is used. It establishes an initial fault coverage based on earlier test sets, reducing the test generation effort.

Phase 2 extends $T(\mathrm{SE}_j, 0)$ by considering target faults that are detected by $T(\mathrm{SE}_0)$ but not by $T(\mathrm{SE}_j, 0)$.

Phase 3 follows the lines of an iterative test compaction procedure that increases the number of faults detected by some tests in order to remove other tests. Phase 3 creates test sets $T(\mathrm{SE}_j, 1), T(\mathrm{SE}_j, 2), \ldots$ for SE_j. Every additional test set increases the fault coverage relative to the previous one. When the fault coverage cannot be increased any further, the final test set is used as $T(\mathrm{SE}_j)$.

Considering all the scan-enable sequences in $\mathrm{SE} = \{\mathrm{SE}_0, \mathrm{SE}_1, \ldots, \mathrm{SE}_{m-1}\}$, the procedure obtains test sets $T(\mathrm{SE}_0)$, $T(\mathrm{SE}_1)$, $T(\mathrm{SE}_2)$, ..., $T(\mathrm{SE}_{m-1})$. For $j \geq 1$, let $\phi(\mathrm{SE}_j)$ be the fault coverage of $T(\mathrm{SE}_j)$. To implement Phase 1, the procedure maintains a set of tests U as follows.

Initially, $U = T(\mathrm{SE}_0)$ and $\phi_{\mathrm{best}} = 0$. For $j \geq 1$, after $T(\mathrm{SE}_j)$ is computed, its fault coverage $\phi(\mathrm{SE}_j)$ is compared with ϕ_{best}. If $\phi(\mathrm{SE}_j) > \phi_{\mathrm{best}}$, the procedure adds $T(\mathrm{SE}_j)$ to U, and updates that $\phi_{\mathrm{best}} = \phi(\mathrm{SE}_j)$. In this way, U contains every test set $T(\mathrm{SE}_j)$ that achieves an increased fault coverage. Experimental results indicate that the tests in these test sets have scan-in states and primary input vectors that are effective when forming tests for other scan-enable

sequences. By storing them in U, they are available as additional scan-enable sequences are considered.

B. Phase 1

For $j \geq 1$, Phase 1 uses the set of tests U to initialize $T(\mathrm{SE}_j)$ by forming a test set denoted by $T(\mathrm{SE}_j, 0)$.

Initially, $T(\mathrm{SE}_j, 0) = \emptyset$, and F includes all the target faults. The tests in U are considered one by one. A test $t_i = \langle s_i, v_i, \mathrm{SE}_i, \mathrm{SI}_i \rangle \in U$ is used for defining a subset of tests for $T(\mathrm{SE}_j, 0)$ as follows.

As before, let the number of 1s in SE_j be u_j. This yields 2^{u_j} options for the scan-in sequence. Let the options be $\mathrm{SI}_{j,0}$, $\mathrm{SI}_{j,1}$, …, $\mathrm{SI}_{j,2^{u_j}-1}$. For every $0 \leq k \leq 2^{u_j} - 1$, the procedure defines the test $\langle s_i, v_i, \mathrm{SE}_j, \mathrm{SI}_{j,k} \rangle$. For example, with $\mathrm{SE}_j = 1100$, the procedure considers the tests

$\langle s_i, v_i, 1100, 0000 \rangle$,
$\langle s_i, v_i, 1100, 0100 \rangle$,
$\langle s_i, v_i, 1100, 1000 \rangle$, and
$\langle s_i, v_i, 1100, 1100 \rangle$.

Fault simulation with fault dropping is carried out for F under every one of the tests. A test that increases the fault coverage is added to $T(\mathrm{SE}_j, 0)$.

C. Phase 2

For $j \geq 1$, Phase 2 adds tests to $T(\mathrm{SE}_j, 0)$ targeting faults in F that are not yet detected. The procedure considers every fault $f \in F$ with up to ten tests from $T(\mathrm{SE}_j, 0)$. For a fault $f \in F$ and a test $t_i = \langle s_i, v_i, \mathrm{SE}_j, \mathrm{SI}_i \rangle \in T(\mathrm{SE}_j, 0)$, the procedure attempts to modify t_i into a test t_{new} that detects f. Initially, $t_{\mathrm{new}} = t_i$.

To guide the modification of t_{new}, the procedure simulates f under t_{new}. It finds the number of state variables for which fault simulation of f yields the values 0/1 or 1/0 under t_{new}. This number is denoted by n_{new}. A higher value of n_{new} makes it more likely that f will be detected.

Using the same notation as before, we have that $t_{\mathrm{new}} = \langle s_{\mathrm{new}}, v_{\mathrm{new}}, \mathrm{SE}_j, \mathrm{SI}_{\mathrm{new}} \rangle$. The modification of t_{new} proceeds by complementing bits of s_{new}, v_{new}, and $\mathrm{SI}_{\mathrm{new}}[u]$ where $\mathrm{SE}_j[u] = 1$. The bits are considered one by one in a random order.

After a bit b is complemented, let the test t^b_{new} be obtained. The procedure simulates f under t^b_{new} and checks whether f is detected. It also finds the number of state variables with 0/1 or 1/0 values, n^b_{new}. If f is detected by t_{new}, the modification of t_{new} terminates. The test is added to $T(\mathrm{SE}_j, 0)$, and the faults it detects (including f) are removed from F.

Otherwise, if $n^b_{\mathrm{new}} \geq n_{\mathrm{new}}$, the complementation is accepted, and the procedure assigns $t_{\mathrm{new}} = t^b_{\mathrm{new}}$ and $n_{\mathrm{new}} = n^b_{\mathrm{new}}$. If $n^b_{\mathrm{new}} < n_{\mathrm{new}}$, the complementation is not accepted, and t^b_{new} is discarded.

If the procedure increases the value of n_{new} for at least one complemented bit, it considers all the bits of t_{new} again.

D. Phase 3

Phase 3 follows the lines of a test compaction procedure that modifies tests to increase the numbers of faults they detect. As it modifies the tests, it also increases the fault coverage. If the fault coverage is already complete, the procedure acts as a test compaction procedure that reduces the number of tests.

Phase 3 is iterative. For $k \geq 1$, it uses $T(\mathrm{SE}_j, k-1)$ to produce $T(\mathrm{SE}_j, k)$ with the same or higher fault coverage. Phase 3 terminates when the fault coverage of $T(\mathrm{SE}_j, k)$ is the same as that of $T(\mathrm{SE}_j, k-1)$.

To produce $T(\mathrm{SE}_j, k)$, the procedure initially assigns $T(\mathrm{SE}_j, k) = \emptyset$, and includes all the target faults in F. In addition, it includes the faults detected by $T(\mathrm{SE}_j, k-1)$ in F_0. The test set $T(\mathrm{SE}_j, k)$ is guaranteed to detect all the faults from F_0 at the end of the iteration. The fault coverage is increased if $T(\mathrm{SE}_j, k)$ also detects faults from $F - F_0$.

The construction of $T(\mathrm{SE}_j, k)$ proceeds by considering the tests from $T(\mathrm{SE}_j, k-1)$ one by one. The tests in $T(\mathrm{SE}_j, k-1)$ are ordered by decreasing number of detected faults. Numbers of detected faults are obtained by fault simulation with fault dropping of $T(\mathrm{SE}_j, k-1)$. This order ensures that more effective tests are considered earlier for the construction of $T(\mathrm{SE}_j, k)$.

When a test $t_i = \langle s_i, v_i, \mathrm{SE}_j, \mathrm{SI}_i \rangle \in T(\mathrm{SE}_j, k-1)$ is considered, the procedure first simulates F under t_i to find the number of detected faults, n_i. If $n_i > 0$, the procedure continues as follows.

If the fault coverage of $T(\mathrm{SE}_j, k)$ is lower than 70% of the fault coverage of $T(\mathrm{SE}_0)$, the procedure accepts t_i as it is. It adds it to $T(\mathrm{SE}_j, k)$, and removes the faults it detects from F. Otherwise, the procedure attempts to improve t_i by increasing the number of faults it detects. This process starts only after the fault coverage of $T(\mathrm{SE}_j, k)$ reaches a threshold to limit the computational effort of improving the tests.

To improve t_i, the procedure complements bits of t_i. The bits that can be complemented are those of s_i, v_i, and $\mathrm{SI}_i[u]$ where $\mathrm{SE}_j[u] = 1$. The bits are considered one by one in a random order.

After a bit b is complemented, let the test t^b_i be obtained. The procedure simulates F under t^b_i and finds the number of detected faults, n^b_i. If $n^b_i \geq n_i$, the complementation is accepted, and the procedure assigns $t_i = t^b_i$ and $n_i = n^b_i$. If $n^b_i < n_i$, the complementation is not accepted, and t^b_i is discarded.

If the procedure increases the number of faults detected by t_i for at least one complemented bit, it considers all the bits of t_i a second time.

While computing n^b_i, the procedure checks whether the test detects any faults from $F - F_0$. In this case, it adds t^b_i to a set denoted by T_{new}. The set T_{new} is emptied before starting the modification of t_i.

After modifying t_i, it is added to $T(\mathrm{SE}_j, k)$, and the faults it detects are removed from F. The procedure then simulates F under the tests in T_{new}. Tests that increase the fault coverage are added to $T(\mathrm{SE}_j, k)$.

All the tests from $T(\mathrm{SE}_j, k-1)$ are considered again as long as additional tests are added to $T(\mathrm{SE}_j, k)$. A test t_i may contribute more than one test to $T(\mathrm{SE}_j, k)$ since the

TABLE IV
TEST GENERATION, EXTENDED SET OF SCAN-ENABLE SEQUENCES

circuit	sv	pi	len	seq	init f.c.	final f.c.	tests	cycles	ntime
s298	14	5	2		0.000	86.242	33	542	1.00
s298	14	5	3	100	85.235	86.242	25	439	53.00
s344	15	11	2		0.000	92.733	25	440	1.00
s344	15	11	3	100	89.099	92.733	22	411	60.00
s382	21	5	2		0.000	81.545	29	688	1.00
s382	21	5	3	100	81.152	81.545	24	597	26.50
s510	6	21	2		0.000	86.863	64	518	1.00
s510	6	21	3	100	86.863	86.863	46	420	15.00
s526	21	5	2		0.000	85.741	67	1562	1.00
s526	21	5	3	100	84.030	85.741	62	1509	28.36
b05	34	4	2		0.000	93.313	67	2446	1.00
b05	34	4	3	100	93.179	93.313	71	2661	65.29
b07	51	4	2		0.000	91.374	61	3284	1.00
b07	51	4	3	100	91.219	91.374	47	2589	53.35
b08	21	12	2		0.000	89.573	58	1355	1.00
b08	21	12	3	100	87.915	89.573	57	1389	44.35
b09	28	4	2		0.000	91.003	27	838	1.00
b09	28	4	3	100	89.381	91.003	25	803	47.50
sasc	117	17	2		0.000	93.636	36	4401	1.00
sasc	117	17	3	100	92.983	93.636	34	4197	142.86
simple_spi	131	17	2		0.000	89.215	63	8510	1.00
simple_spi	131	17	3	100	88.770	89.215	54	7367	188.14
systemcdes	190	132	2		0.000	96.404	81	15742	1.00
systemcdes	190	132	3	100	96.355	96.404	60	11770	176.41
usb_phy	98	16	2		0.000	94.872	39	3998	1.00
usb_phy	98	16	3	100	93.797	94.872	37	3835	293.25
b03	30	7	2		0.000	90.885	29	958	1.00
b03	30	7	4	0010	90.234	90.885	26	914	64.90
s1423	74	19	2		0.000	85.278	58	4482	1.00
s1423	74	19	4	1001	85.137	85.278	44	3506	98.55
b10	17	14	2		0.000	84.598	44	853	1.00
b10	17	14	5	00110	82.069	84.598	45	1007	140.17
b11	30	10	2		0.000	94.645	80	2590	1.00
b11	30	10	5	00110	92.951	94.645	70	2480	177.22
b04	66	14	2		0.000	80.823	39	2718	1.00
b04	66	14	5	10100	80.604	80.823	28	2054	70.81
steppermotordrive	25	5	2		0.000	96.164	45	1240	1.00
steppermotordrive	25	5	5	11000	83.995	96.164	39	1195	233.27
s208	8	13	2		0.000	82.692	35	358	1.00
s208	8	13	8	00111110	82.692	82.692	24	392	86.00
s420	16	21	2		0.000	85.238	57	1042	1.00
s420	16	21	10	1011111100	84.643	84.643	36	952	383.50

modification may cause the test to detect different faults in different passes over $T(\mathrm{SE}_j, k-1)$.

Considering the ability of the procedure to achieve test compaction, the following points are important. As long as the procedure finds tests to detect faults from $F - F_0$, it adds them to T_{new} and then to $T(\mathrm{SE}_j, k)$. These tests may detect single faults, and the procedure adds them to $T(\mathrm{SE}_j, k)$ without further modification. As a result, the test set is not compact. However, in the next iteration, the same tests are modified to increase the numbers of faults they detect.

The procedure does not terminate as long as $T_{\mathrm{new}} \neq \emptyset$ is obtained for any test since this increases its fault coverage. It only terminates when the fault coverage of $T(\mathrm{SE}_j, k)$ is the same as that of $T(\mathrm{SE}_j, k-1)$. At this point, tests have been modified to increase their numbers of detected faults. Therefore, the test set is compact.

In addition, after the last iteration, the procedure applies forward-looking reverse order fault simulation to remove unnecessary tests. The final test set is assigned to $T(\mathrm{SE}_j)$.

E. Experimental Results

The test generation procedure was applied to several larger benchmark circuits. Additional circuits are considered in Section VI.

The results are reported in Table IV in a format similar to Tables II and III. In addition to the columns of Tables II and III, column $init\ f.c.$ of Table IV shows the fault coverage of $T(\mathrm{SE}_j, 0)$ obtained based on the set of tests U, and column $final\ f.c.$ shows the final fault coverage of $T(\mathrm{SE}_j)$. Column $ntime$ shows the normalized test generation time, where the runtime for test generation is divided by the runtime for fault simulation with fault dropping of $T(\mathrm{SE}_0)$.

There is only one circuit in Table IV, $s420$, for which the transition fault coverage achieved with a single scan-enable sequence is lower than that achieved with broadside and skewed-load tests. The single scan-enable sequence may still be used since it simplifies the test application process and the fault coverage is higher than that achieved using only

TABLE V
REDUCED SET OF SCAN-ENABLE SEQUENCES

λ_j	sequences	λ_j	sequences
3	100	4	-
5	10010	6	100110
	10100		101100
7	1000110	8	10001110
	1001100		10011100
	1001110		10011110
	1011000		10111000
	1011100		10111100
9	100001110	10	1000011110
	100011100		1000111100
	100011110		1000111110
	100111000		1001111000
	100111100		1001111100
	100111110		1001111110
	101110000		1011110000
	101111000		1011111000
	101111100		1011111100

broadside or only skewed-load tests, which is the common practice.

In general, the results in Table IV are consistent with the results in Tables II and III. Specifically, there is a significant number of circuits for which a scan-enable sequence SE_j that is longer than three allows $T(\mathrm{SE}_j)$ to achieve the fault coverage of $T(\mathrm{SE}_0)$.

As in Tables II and III, the same scan-enable sequences are effective for different circuits. Moreover, most of the sequences in Tables II–IV have common properties that are discussed in Section VI.

The initial fault coverage achieved by $T(\mathrm{SE}_j, 0)$ is significant in all the cases. As additional scan-enable sequences are considered, the initial fault coverage of $T(\mathrm{SE}_j, 0)$ typically increases. Thus, incremental test generation is effective in this experiment. As a result, the normalized runtime does not increase linearly with the length of the scan-enable sequence even though linearly more clock cycles need to be simulated for every test.

The number of tests is reduced similar to Tables II and III. Since the circuits have larger numbers of state variables, scan operations have a larger contribution to the number of clock cycles. With fewer tests, and the same fault coverage, there are fewer scan operations, and the number of clock cycles for test application is typically reduced. Further reductions can be obtained by modifying more of the tests in the final test set during the last iteration of Phase 3.

VI. REDUCED SET OF SCAN-ENABLE SEQUENCES

This section discusses properties of the scan-enable sequences selected in Tables II–IV. These properties are used for reducing the number of sequences in SE, thus limiting the number of sequences for which test generation will be carried out when other circuits are considered.

In addition, the results in Table IV indicate that when the test generation procedure is used, Phases 1 and 2 have a larger contribution to the fault coverage than Phase 3. Moreover, Phase 3 does not typically compensate for a lower fault coverage achieved after Phases 1 and 2. Therefore, the selection of a scan-enable sequence from SE can be done based on the fault coverage achieved after Phases 1 and 2. Phase 3 is applied only to the selected scan-enable sequence to increase the fault coverage further and achieve test compaction.

TABLE VI
EXHAUSTIVE TEST SETS, REDUCED SET OF SCAN-ENABLE SEQUENCES

circuit	sv	pi	len	seq	f.c.	tests	cycles
mc	2	5	2		81.538	13	54
mc	2	5	3	100	81.538	9	47
lion	2	4	2		64.706	6	26
lion	2	4	5	10010	64.706	4	30
train4	2	4	2		69.118	4	18
train4	2	4	5	10010	69.118	3	23
beecount	3	5	2		62.871	15	78
beecount	3	5	5	10010	62.871	11	91
bbtas	3	4	2		75.893	14	73
bbtas	3	4	5	10010	75.893	8	67
lion9	3	4	2		71.818	12	63
lion9	3	4	5	10100	71.818	10	83
train11	4	4	2		71.978	21	130
train11	4	4	6	100110	71.978	12	124
firstex	4	4	2		70.290	20	124
firstex	4	4	7	1001110	70.290	12	136
keyb	5	9	2		72.041	84	593
keyb	5	9	7	1001110	72.041	56	677
ex2	5	4	2		82.581	54	383
ex2	5	4	8	10011110	82.581	38	499
donfile	5	4	2		77.208	61	432
donfile	5	4	8	10111100	77.208	48	629

A. Properties

Property 1: Most of the sequences in Tables II–IV have a run of two or more consecutive 0s, corresponding to two or more consecutive functional capture cycles. This is also a property of a broadside test.

Property 2: Most of the sequences end with a functional capture cycle. This is important since the functional capture cycle allows fault effects to be latched in the flip-flops before the scan-out operation at the end of the test.

Property 3: Most of the sequences have either one or two runs of consecutive 1s, corresponding to scan shift cycles. Thus, a sequence such as 101010100, where single scan shift and functional capture cycles alternate, is not effective for the circuits considered.

Property 4: In most of the sequences, the number of scan shift cycles u_j is such that $u_j \geq \lambda_j/2$.

Property 5: Many of the sequences in Tables II–IV start with $\mathrm{SE}_j(0)\mathrm{SE}_j(1) = 10$. This is also a property of a skewed-load test. Although there are exceptions where the sequence starts with $\mathrm{SE}_j(0) = 0$ or $\mathrm{SE}_j(0)\mathrm{SE}_j(1) = 11$, additional experimental results indicate that this property typically holds.

B. Reduced Set

The set SE of possible scan-enable sequences was reduced based on Properties 1–5. The scan-enable sequences for $3 \leq \lambda_j \leq 10$ are shown in Table V. The extended set of scan-enable sequences contains $2^{\lambda_j} - 2$ sequences of length λ_j. The reduced set contains one sequence of length $\lambda_j = 3$, no sequences of length $\lambda_j = 4$, two sequences of length $\lambda_j = 5$, and so on.

Using the reduced set SE, Phases 1 and 2 of the test generation procedure are applied to all the scan-enable sequences in SE. Of all the sequences in SE, the one for which Phases 1 and 2 yield the highest fault coverage is selected. The entire test generation procedure, including Phase 3, is then applied to the selected scan-enable sequence.

C. Experimental Results

The results obtained with the reduced set of scan-enable sequences are shown in Tables VI and VII. Table VI shows circuits from Tables II and III for which exhaustive test sets are used. The circuits are such that the scan-enable sequences selected for them from the extended set are not included in the reduced set. Therefore, it is interesting to consider these circuits again.

The first row for every circuit in Table VI shows the results using $\mathrm{SE}_0 = 00, 10$ (or a broadside and skewed-load test set). The second row shows the results obtained for the scan-enable sequence SE_j selected from the reduced set SE.

Table VII shows larger circuits for which test generation is applied. The circuits in the first part of Table VII also appear in Table IV, but the scan-enable sequences selected for them in Table IV are not included in the reduced set. The second and third parts of Table VII show additional larger benchmark circuits that were not considered in Table IV.

The first row for every circuit in Table VII shows the results using $\mathrm{SE}_0 = 00, 10$. The second row shows the results obtained for the selected scan-enable sequence $\mathrm{SE}_j \in SE$ when only Phases 1 and 2 are applied. The final fault coverage in this case is that obtained by Phases 1 and 2. The third row, when it is included, shows the results obtained for the selected scan-enable sequence $\mathrm{SE}_j \in \mathrm{SE}$ when the entire test generation procedure is applied.

The results in Tables VI and VII further demonstrate the possibility of using a single scan-enable sequence to replace broadside and skewed-load tests. For the circuits from Tables II–IV, a different scan-enable sequence is selected from the reduced set, but the results are similar to those obtained with the extended set.

In general, the fault coverage obtained in Table VII is close to that obtained when both test types are used. This fault coverage is higher than that obtained when only one of the test types is used, which is the common practice. In addition, the selected scan-enable sequence has the advantage of a simplified test application process with a single test type represented by a single scan-enable sequence.

With an upper bound of $\lambda_j = 10$ on the length of a scan-enable sequence, the test generation procedure selects a sequence SE_j with $\lambda_j = 10$ in a small number of cases. Consequently, longer sequences are not likely to be useful.

The normalized runtime depends more strongly on the length of the scan-enable sequence than on the size of the circuit. This indicates that the procedure scales similar to a fault simulation procedure.

Finally, to demonstrate feasibility for larger circuits, Table VIII shows results obtained for industrial designs. A commercial tool was used for test generation in this case. The tool has several restrictions on the generated tests: 1) it holds primary input vectors constant during a test; 2) it does not allow the scan vector to change between the scan operations of a test; and 3) it does not observe primary output values. These restrictions limit the achievable fault coverage, affecting the longer scan-enable sequences more significantly. Specifically, holding the scan vector between scan operations affects more severely the longer sequences that have more scan shift cycles. In addition, not observing primary output values limits the effectiveness of a longer scan-enable sequence. Therefore, we only report the results for $\mathrm{SE}_j = 100$ in Table VIII to demonstrate the feasibility of the approach suggested in this article when considering industrial designs.

In Table VIII, column *gates* shows the number of gates. Column $brd + skw$ shows information about the test set that consists of broadside and skewed-load tests. Column brd shows information about the test set that consists of broadside tests only. Column skw shows information about the test set that consists of skewed-load tests only. Column SE_j shows information about the test set $T(\mathrm{SE}_j)$. In every case, subcolumn $f.c.$ shows the transition fault coverage, and subcolumn *tests* shows the number of tests. In the case of $T(\mathrm{SE}_j)$, subcolumn *len* shows the length of SE_j, and subcolumn *seq* shows the sequence SE_j.

Table VIII demonstrates that, using a single scan-enable sequence, the fault coverage is close to that achieved by broadside and skewed-load tests, and higher than that achieved by a single test type. Thus, the single scan-enable sequence is preferred over using only one test type. More importantly, the loss in fault coverage compared with the case where both test types are used is similar to that obtained in Table VII for benchmark circuits, and it does not grow with the size of the circuit.

D. Diagnostic Fault Simulation

Next, we consider the effects of using a single scan-enable sequence on the diagnosis quality.

The ability of a test set to support fault diagnosis can be measured by performing diagnostic fault simulation. Diagnostic fault simulation produces a set of equivalence classes. Two faults are included in the same equivalence class if they are not distinguished by the test set. In this case, a fault diagnosis procedure will not be able to determine that one of the faults, but not the other, is likely to be present in a faulty circuit. Smaller equivalence classes, and fewer indistinguished fault pairs, imply a better diagnosis quality.

The results of diagnostic fault simulation are reported in Table IX for several of the circuits from Table VII. The test sets compared in Table IX are the test set $T_{\mathrm{brd+skw}}$ that consists of both broadside and skewed-load tests, and the test set $T(\mathrm{SE}_j)$ selected in Table VII. Only faults that are detected by both test sets are used for diagnostic fault simulation.

TABLE VII
TEST GENERATION, REDUCED SET OF SCAN-ENABLE SEQUENCES

circuit	sv	pi	len	seq	init f.c.	final f.c.	tests	cycles	ntime
b03	30	7	2		0.000	90.885	29	958	1.00
b03	30	7	6	101100	89.453	90.885	41	1506	28.00
b03	30	7	6	101100	90.885	90.885	18	678	65.00
steppermotordrive	25	5	2		0.000	96.164	45	1240	1.00
steppermotordrive	25	5	7	1001100	95.635	95.635	56	1817	20.25
steppermotordrive	25	5	7	1001100	95.635	95.635	37	1209	79.75
s208	8	13	2		0.000	82.692	35	358	1.00
s208	8	13	8	10111100	82.692	82.692	43	696	29.00
s208	8	13	8	10111100	82.692	82.692	18	296	56.00
b11	30	10	2		0.000	94.645	80	2590	1.00
b11	30	10	9	101110000	94.262	94.317	102	4008	60.75
b11	30	10	9	101110000	94.317	94.645	54	2136	233.52
s1423	74	19	2		0.000	85.278	58	4482	1.00
s1423	74	19	10	1001111110	84.996	85.067	105	8894	596.11
s1423	74	19	10	1001111110	85.067	85.278	40	3434	988.07
b10	17	14	2		0.000	84.598	44	853	1.00
b10	17	14	10	1011110000	84.483	84.598	69	1880	101.00
b10	17	14	10	1011110000	84.598	84.598	32	881	163.17
s35932	1728	37	2		0.000	73.492	29	51898	1.00
s35932	1728	37	3	100	73.448	73.492	46	81354	0.62
s35932	1728	37	3	100	73.492	73.492	22	39810	248.68
wb_dma	523	217	2		0.000	83.233	177	93448	1.00
wb_dma	523	217	3	100	80.569	81.856	175	92573	12.65
wb_dma	523	217	3	100	81.856	83.215	114	60487	815.00
i2c	128	19	2		0.000	81.841	72	9488	1.00
i2c	128	19	5	10010	79.627	80.676	120	16088	12.14
i2c	128	19	5	10010	80.676	81.702	54	7310	224.77
spi	229	47	2		0.000	91.009	494	114343	1.00
spi	229	47	7	1001100	90.541	90.600	476	112565	18.25
spi	229	47	7	1001100	90.600	91.001	261	61825	1434.46
s15850	597	16	2		0.000	90.437	328	197069	1.00
s15850	597	16	9	100111110	88.859	89.159	403	244815	331.15
s15850	597	16	9	100111110	89.159	90.361	154	93921	4021.73
b14	247	35	2		0.000	84.162	219	54778	1.00
b14	247	35	9	101110000	81.781	81.991	282	72439	125.32
b14	247	35	9	101110000	81.991	84.162	144	37111	1858.03
s5378	179	37	2		0.000	83.513	233	42352	1.00
s5378	179	37	10	1001111110	82.068	82.115	310	58769	950.71
s5378	179	37	10	1001111110	82.115	83.362	151	28718	2630.89
systemcaes	670	260	2		0.000	95.602	160	108190	1.00
systemcaes	670	260	10	1001111110	95.448	95.529	390	265870	591.47
systemcaes	670	260	10	1001111110	95.529	95.599	94	64590	4572.79
s38584	1452	14	2		0.000	82.192	473	689194	1.00
s38584	1452	14	3	100	80.130	80.600	552	804612	7.99
aes_core	530	260	2		0.000	97.452	275	146830	1.00
aes_core	530	260	7	1000110	97.452	97.452	660	354950	10.55
s13207	669	33	2		0.000	94.859	445	299264	1.00
s13207	669	33	8	10001110	91.885	92.351	481	326306	213.52
s38417	1636	30	2		0.000	98.254	663	1087630	1.00
s38417	1636	30	8	10001110	95.536	95.874	848	1395748	395.24
tv80	359	15	2		0.000	96.330	706	255225	1.00
tv80	359	15	9	100001110	92.590	93.086	758	279303	236.95
b20	494	35	2		0.000	88.544	288	143342	1.00
b20	494	35	9	100111100	86.863	87.047	453	228353	149.29
s9234	228	21	2		0.000	85.229	275	63478	1.00
s9234	228	21	9	101110000	80.198	81.075	500	118728	298.98
b15	447	38	2		0.000	93.715	440	198007	1.00
b15	447	38	10	1001111110	92.362	92.442	787	360106	1339.23

TABLE VIII
RESULTS FOR INDUSTRIAL CIRCUITS

		brd+skw		brd		skw		SE_j			
circuit	gates	f.c.	tests	f.c.	tests	f.c.	tests	len	seq	f.c.	tests
designA	1.27M	94.75	5115	81.71	5464	92.15	4187	3	100	93.71	7862
designB	4.22M	92.92	2686	85.52	2504	91.38	3415	3	100	92.69	2600

Column $f.c.$ of Table IX shows the percentage of faults detected by both test sets. For each test set, subcolumn *tests* shows the number of tests. Subcolumn *len* shows the length of the scan-enable sequence. Subcolumn *class*

TABLE IX
RESULTS OF DIAGNOSTIC FAULT SIMULATION

		brd+skw				SE_j			
circuit	f.c.	tests	len	class	pairs	tests	len	class	pairs
b03	90.885	29	2	6	227	18	6	5	164
steppermotordrive	95.635	45	2	5	259	37	7	5	226
s208	82.692	35	2	9	352	18	8	6	241
b11	94.645	80	2	8	528	54	9	7	387
s1423	85.278	58	2	10	1000	40	10	10	771
b10	84.598	44	2	6	218	32	10	5	203
s35932	73.492	29	2	11	46340	22	3	11	44020
wb_dma	83.215	177	2	12	4607	114	3	17	4410
i2c	81.702	72	2	8	1619	54	5	9	1444
spi	91.001	494	2	12	4803	261	7	8	3589
s15850	90.361	328	2	98	48570	154	9	98	47508
b14	84.162	219	2	107	14858	144	9	39	4890
s5378	83.362	233	2	17	7867	151	10	23	7333
systemcaes	95.599	160	2	6	7272	94	10	5	5983

shows the number of faults in the largest equivalence class. Subcolumn *pairs* shows the number of indistinguished fault pairs.

From Table IX, it can be observed that the number of faults in the largest equivalence class is typically smaller for $T(\mathrm{SE}_j)$. The number of indistinguished fault pairs is also smaller for $T(\mathrm{SE}_j)$. Thus, the diagnosis quality does not deteriorate, and even improves, when $T(\mathrm{SE}_j)$ is used instead of $T_{\mathrm{brd+skw}}$.

VII. CONCLUSION

Transition fault test sets that consist of both broadside and skewed-load tests achieve higher fault coverage and better test compaction than a test set that contains only one of the test types. Broadside and skewed-load tests differ in the scan-enable sequences they use. This article considered the problem of generating a complete test set for transition faults where all the tests use the same scan-enable sequence. This simplifies the test application process that does not need to accommodate different test types or multiple scan-enable sequences. This article studied this issue using exhaustive test sets as well as test generation. Experimental results for benchmark circuits demonstrated that there are significant numbers of circuits for which a single scan-enable sequence is sufficient. In many cases, the sequence length is larger than three. The sequences for different circuits have common features that can be used for limiting the number of different sequences that need to be considered for a circuit.

REFERENCES

[1] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 8, pp. 1232–1241, Aug. 1993.

[2] J. Savir and S. Patil, "Broad-side delay test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 8, pp. 1057–1064, Aug. 1994.

[3] N. Ahmed, M. Tehranipoor, C. P. Ravikumar, and K. M. Butler, "Local at-speed scan enable generation for transition fault testing using low-cost testers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 5, pp. 896–906, May 2007.

[4] G. Xu and A. D. Singh, "Scan cell design for launch-on-shift delay tests with slow scan enable," *IET Comput. Digit. Techn.*, vol. 1, pp. 213–219, May 2007.

[5] I. Pomeranz, "Static test compaction for delay fault test sets consisting of broadside and skewed-load tests," in *Proc. 29th VLSI Test Symp.*, May 2011, pp. 84–89.

[6] L. N. Reddy, I. Pomeranz, and S. M. Reddy, "COMPACTEST-II: A method to generate compact two-pattern test sets for combinational logic circuits," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 1992, pp. 568–574.

[7] R. Desineni, K. N. Dwarkanath, and R. D. Blanton, "Universal test generation using fault tuples," in *Proc. Int. Test Conf.*, Oct. 2000, pp. 812–819.

[8] G. Chen, S. Reddy, I. Pomeranz, J. Rajski, P. Engelke, and B. Becker, "A unified fault model and test generation procedure for interconnect opens and bridges," in *Proc. Eur. Test Symp.*, May 2005, pp. 22–27.

[9] S. Goel and R. A. Parekhji, "Choosing the right mix of at-speed structural test patterns: Comparisons in pattern volume reduction and fault detection efficiency," in *Proc. 14th Asian Test Symp. (ATS)*, 2005, pp. 330–336.

[10] S. Alampally, R. T. Venkatesh, P. Shanmugasundaram, R. A. Parekhji, and V. D. Agrawal, "An efficient test data reduction technique through dynamic pattern mixing across multiple fault models," in *Proc. 29th VLSI Test Symp.*, May 2011, pp. 285–290.

[11] C.-H. Wu and K.-J. Lee, "Transformation of multiple fault models to a unified model for ATPG efficiency enhancement," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2016, pp. 1–10.

[12] I. Pomeranz, "Skewed-load tests for transition and stuck-at faults," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 10, pp. 1969–1973, Oct. 2019.

[13] I. Pomeranz and S. M. Reddy, "Transparent scan: A new approach to test generation and test compaction for scan circuits that incorporates limited scan operations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 12, pp. 1663–1670, Dec. 2003.

[14] I. Park and E. J. McCluskey, "Launch-on-shift-capture transition tests," in *Proc. IEEE Int. Test Conf.*, Oct. 2008, pp. 1–9.

[15] I. Pomeranz, "Test compaction under bounded transparent-scan," in *Proc. IEEE 37th VLSI Test Symp. (VTS)*, Apr. 2019, pp. 1–6.

[16] S. Y. Lee and K. K. Saluja, "Test application time reduction for sequential circuits with scan," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 9, pp. 1128–1140, Sep. 1995.

[17] X. Lin and R. Thompson, "Test generation for designs with multiple clocks," in *Proc. Design Autom. Conf.*, Jun. 2003, pp. 662–667.

[18] G. Bhargava, D. Meehl, and J. Sage, "Achieving serendipitous N-detect mark-offs in multi-capture-clock scan patterns," in *Proc. IEEE Int. Test Conf.*, Oct. 2007, pp. 1–7, paper 30.2.

[19] T. McLaurin and I. P. Lawrence, "Improving power, performance and area with test: A case study," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2018, pp. 1–10.

[20] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-in Test for VLSI Pseudorandom Techniques*. Hoboken, NJ, USA: Wiley Interscience, 1987.