Compact Test Generation With an Influence Input Measure for Launch-On-Capture Transition Fault Testing

Dong Xiang, *Senior Member, IEEE*, Wenjie Sui, Boxue Yin, and Kwang-Ting Cheng, *Fellow, IEEE*

***Abstract*— We propose a compact test generation method for transition faults based on a conflict avoidance driven scheme. A new measure is proposed to estimate the influence inputs, which is the subset of inputs to be specified, needed for detecting transition faults. The value requirements at the pseudoprimary inputs (PPIs) of the second frame of the automatic test pattern generation circuit model are partitioned into separate subsets. The sequential backtracing scheme backtraces the value requirements on the PPIs of the second frame subset-by-subset. Justification of the necessary value requirements at the PPIs in the second frame is completed using a conflict-driven procedure. With an influence input measure for transition faults under the launch-on-capture scan testing application scheme, a new dynamic test compaction scheme is proposed. The new test compaction scheme tries to compact as many faults as possible into the current test. Experimental results and comparison with existing approaches demonstrate the efficiency and effectiveness of the proposed method.**

***Index Terms*— At-speed testing, launch-on-capture (LOC) delay testing, test data compression, test response compaction.**

I. Introduction

TEST generation for stuck-at faults in combinational circuits tried to generate a minimal test set to cover all testable faults in as less as possible CPU time. In the earlier PODEM [8], FAN [7], SOCRATES [28] algorithms, the recursive learning test generation scheme [14] solved CPU time and complete coverage problems very well. Techniques to reduce test set size for single stuck-at faults, called test compaction, were proposed by methods later [10]. Another way to reduce test data volume is to compress test stimuli and compact responses [11]. We had some special techniques to compress test data and compact test responses for launch-on-capture (LOC) delay testing [3], [5], [34] using novel scan architectures and new test application schemes.

Delay testing can detect performance-related defects that might not be discovered by single stuck-at fault tests [12], [30], [37]. Detecting a path delay fault or a transition fault requires the application of a vector pair to the combinational portion of the circuit under test. In [6], effective algorithms for synchronous sequential circuits were proposed to generate delay tests and fault simulate them. Depending on how the second vector of the test pair is produced, there are two different ways of applying delay tests for a standard scan design: 1) LOC testing [27] and 2) skewed load. In [16], Lin, Lu, and Cheng addressed the potential over-testing problem of ac scan and proposed adding extra functional constraints for the circuit under test when generating delay tests. These additional constraints allow this testing method to operate much like functional testing. Such tests are called pseudofunctional tests. Efficient techniques were introduced to automatically identify useful constraints for generating pseudofunctional tests.

LOC tests alone usually cannot achieve sufficient fault coverage. Several methods have been proposed to improve delay fault coverage for circuits using a standard scan implementation. Mao and Ciletti [21] suggested a heuristic scan chain ordering for skewed-load delay fault testing so that a maximal number of vector pairs for delay faults can be applied robustly. Liou *et al.* [18] proposed a two-phased multiple-clocked scheme to improve delay fault testing. Lin *et al.* [17] proposed new techniques for at-speed scan testing that can be applied using internal phase-locked loops. Liou, Krstic, Jiang, and Cheng in [19] incorporated statistical information into timing analysis to compute the performance sensitivity of internal signals subject to a given type of defect, noise, or variation sources. They proposed a new path or path segment selection methodology for delay testing based on the results of statistical performance sensitivity analysis. Ahmed *et al.* [1] proposed hybrid scan testing methods by selecting a small set of scan flip-flops as skewed-load scan flip-flops, and the remainder were tested using the LOC test method.

Chen *et al.* [3] proposed a new scan architecture and a test application scheme to improve transition fault coverage by selecting a small number of scan flip-flops as either skewed-load or enhanced scan flip-flops, which helps to reduce the power requirements as well. Yin *et al.* [36] proposed techniques to improve the fault coverage of small delay defects and the test compaction effectiveness based on structural analysis by selecting the first scan flip-flops of the scan chains as enhanced flip-flops. The values of these selected enhanced scan flip-flops that correspond to the stable vector can be applied directly if no hold latch is required. Chen *et al.* [5] proposed a new flip-flop partitioning scheme to reduce capture power for at-speed LOC delay testing.

Manuscript received July 22, 2012; revised April 24, 2013; accepted August 20, 2013. Date of publication September 25, 2013; date of current version August 21, 2014. This work was supported in part by the National Science Foundation of China under Grant 60910003, Grant 61170063, and Grant 61373021, and in part by the project from the Education Ministry under Grant 20111081042.

D. Xiang, W. Sui, and B. Yin are with the School of Software, Tsinghua University, Beijing 100084, China (e-mail: dxiang@tsinghua.edu.cn; swj1212qw@163.com).

K.-T. Cheng is with the Department of Electrical Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: timcheng@ece.ucsb.edu).

Digital Object Identifier 10.1109/TVLSI.2013.2280170

The test data generated for detecting delay faults are usually much greater than that of stuck-at faults. Test compaction attempts to reduce the number of test vectors and the size of the test data, which is one of the limiting factors for delay fault testing. Test compaction techniques can be classified into two general categories: 1) static test compaction and 2) dynamic test compaction. Static methods compact tests after the test set has been generated by excluding some test vectors from the test set, fault detected by which can be covered by one or more other test vectors. Dynamic methods conduct test compaction during the test generation process. Another volume reduction technique is to eliminate or reduce the required number of bits that must be kept by the ATE, which is called test data compression [11], [34]. We still do not have sufficient research in the area of test data compression for at-speed delay fault testing.

Compact test generation methods for mixed LOC and skewed-load transition fault testing were proposed in [24] and [25]. Multicycle test compaction for LOC transition fault test generation [23] provides high-quality tests that are closer to functional or pseudofunctional tests [16]. Wang and Walker [31] proposed a dynamic test compaction scheme that covers the K longest paths passing each gate. The method in Wen, *et al.* [32] generates compact tests for transition faults with little capture power by properly filling the "don't care". Hamzaoglu and Patel [9] proposed an effective hybrid method by combining static and dynamic test compaction. Xiang *et al.* [33] proposed a compact test generation scheme for path delay faults using a fast and accurate fault simulation method. The dynamic test compaction scheme in [33] tried to generate compact test pairs, each of which covers as many as possible path delay faults based on output cones. Most recently, a novel path selection procedure and test compaction technique were proposed in [35] for small delay defect testing. The path selection procedure does not need any backtracking. An effective test compaction algorithm for single stuck-at faults, called essential fault reduction, was presented in [10]. Static test compaction combined with some efficient dynamic test compaction heuristics established the compact automatic test pattern generation (ATPG) tool MinTest.

In this paper, we propose a new dynamic test compaction method. The dynamic test compaction scheme uses a new measurement that estimates the influence inputs of a transition fault, the subset of inputs that can be specified to detect the transition fault. It attempts to compact tests for as many faults as possible into the current test.

In the rest of this paper, we introduce a new measure in Section II to estimate the influence inputs of a transition fault. The influence inputs for a transition fault are estimated as the subset of inputs required specifying to generate a test pair for the fault. The test generation process is completed under the guidance of the influence input measure in Section III. A new scheme is proposed to define the input dependency graph (IDG) for guiding justification. We then present a new dynamic test compaction scheme based on the influence input measure in Section IV. A simple and new static test compaction scheme is also proposed, which effectively reduces the pattern count and CPU time. Experimental results are presented in Section V, followed by the conclusion in Section VI.

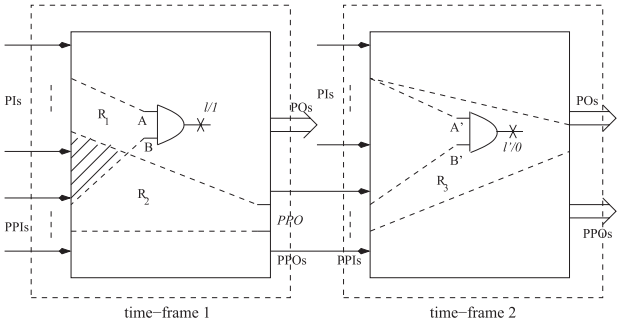

Fig. 1. Two frame model for generating a LOC scan test pair for the rising transition at node l.

II. Influence Input Measure for Transition Fault Testability

The two time frame circuit model for the rising transition fault l/r at node l is shown in Fig. 1. Through generating the test for the fault, the stuck-at-1 fault is activated in the first time frame. In the second time frame, the $l/0$ fault is activated and the subsequent fault effect is propagated to either a primary output (PO) or pseudoprimary output (PPO).

The influence inputs of a transition fault are defined as the subset of primary inputs (PIs) or pseudoprimary inputs (PPIs) that need to be specified to detect a fault. The influence inputs of a transition fault can be estimated based solely on structural analysis in the two frame circuit as shown in Fig. 1. Such an estimation, called the structural influence inputs $IC(l)$ of a node l, can be derived for a transition fault as follows: the POs and PPOs that are reachable from the fault location in the second time frame—denoted as the POs—are marked. The PIs or PPIs in the first time frame are reached by tracing backward from the outputs in the POs and from the fault location in the first time frame, which is denoted as $IC(l)$. These form the structural influence inputs for the fault at l.

The observation reachability function $RO(l)$ is defined as the smallest set of PIs and PPIs that need to be specified for propagating the fault effect at l to either a PO or a PPO. The i-control reachability $RC_i(l)$ is defined as the minimum set of PIs and PPIs that has to be specified to set the value of i to line $l (i \in \{0, 1\})$.

Each PI in the second time frame can be considered as a different PI from the corresponding one in the first time frame, where the control reachability function of a PI in both time frames contains only itself. A PO in the second time frame is treated as a different PO from the corresponding one in the first time frame. The observation reachability function of a PPO in the second time frame is empty because test responses at the PPOs can be completely observed. It is not necessary to calculate the observation reachability function of nodes in the first time frame, since this data will not be used by either the test pattern generation or compaction procedures.

The new measure is introduced to estimate the influence inputs of a transition fault in LOC delay testing. Let l be a PI or PPI in the first time frame. Equations (1)–(12) present

the controllability of all nodes in the first time frame. Let $l(0)$, $l(1)$, and $l(b)$ represent the PI or PPI to be set to value 0, 1, and both values 0 and 1. We have

$$RC_1(l) = \{l(1)\} \qquad RC_0(l) = \{l(0)\}. \tag{1}$$

For an AND gate l with inputs A and B, it is

$$RC_1(l) = RC_1(A) \cup RC_1(B). \tag{2}$$

When a conflict occurs at an input I [i.e., both $I(0)$ and $I(1)$ are present in a control reachability function], replace $I(0)$ and $I(1)$ by $I(b)$ to obtain the control reachability function. For $i \in \{0, 1\}$

$$\begin{aligned} \{I(1)\} \cup \{I(0)\} &= \{I(b)\} \\ \{I(b)\} \cup \{I(i)\} &= \{I(b)\}. \end{aligned} \tag{3}$$

The 0-control reachability function and 0-controllability measure of the output l of a 2-input AND gate with inputs A and B can be calculated as follows:

$$RC_0(l) = \begin{cases} RC_0(A), & \text{if} |RC_0(A)| \le |RC_0(B)| \\ RC_0(B), & \text{if} |RC_0(A)| > |RC_0(B)|. \end{cases} \tag{4}$$

The 1-control reachability function for the output l of a 2-input NAND gate with inputs A and B can be calculated as follows:

$$RC_0(l) = RC_1(A) \cup RC_1(B) \tag{5}$$

$$RC_1(l) = \begin{cases} RC_0(A), & \text{if} |RC_0(A)| \le |RC_0(B)| \\ RC_0(B), & \text{if} |RC_0(A)| > |RC_0(B)|. \end{cases} \tag{6}$$

For an OR gate l with inputs A and B, the influence input function is derived by

$$RC_0(l) = RC_0(A) \cup RC_0(B). \tag{7}$$

The 1-control reachability function at the output l is determined as follows:

$$RC_1(l) = \begin{cases} RC_1(A), & \text{if} |RC_1(A)| \le |RC_1(B)| \\ RC_1(B), & \text{if} |RC_1(A)| > |RC_1(B)|. \end{cases} \tag{8}$$

For a fan-out stem s with branches B_1, B_2, ..., B_k, for $i \in \{0, 1\}$ and $j \in \{1, 2, \ldots, k\}$, we have

$$RC_i(B_j) = RC_i(s). \tag{9}$$

For an inverter l with an input A, the equations used are

$$RC_1(l) = RC_0(A) \qquad RC_0(l) = R_1(A). \tag{10}$$

Let A and B be the inputs of a XOR gate l. We then have

$$\begin{aligned} P_1 &= |RC_1(A) \cup RC_0(B)| \\ P_2 &= |RC_1(B) \cup RC_0(A)| \\ P_3 &= |RC_1(A) \cup RC_1(B)| \\ P_4 &= |RC_0(A) \cup RC_0(B)| \end{aligned}$$

$$RC_1(l) = \begin{cases} RC_1(A) \cup RC_0(B), & \text{if } P_1 \le P_2 \\ RC_1(B) \cup RC_0(A), & \text{otherwise} \end{cases} \tag{11}$$

$$RC_0(l) = \begin{cases} RC_0(A) \cup RC_0(B), & \text{if } P_4 \le P_3 \\ RC_1(B) \cup RC_1(A), & \text{otherwise.} \end{cases} \tag{12}$$

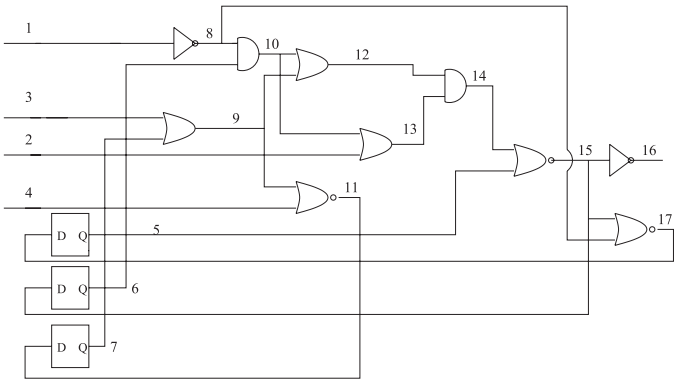

Fig. 2. Example circuit.

The control reachability function for NOR, or NXOR gates can be calculated similarly. The observation reachability function $RO(l)$ is defined as the smallest set of PIs and PPIs that need to be specified to propagate the fault effect at l either to a PO or PPO. We have $RO(l) = \emptyset$ for a PO or a PPO l. Equation (13) through (15) present the calculations of the observation reachability functions. Consider an AND gate l with inputs A and B

$$RO(A) = RO(l) \cup RC_1(B). \tag{13}$$

Replace $I(0)$ and $I(1)$ by $I(b)$ when both $I(0)$ and $I(1)$ occur. Let l be the output of an OR gate with inputs A and B

$$RO(A) = RO(l) \cup RC_0(B). \tag{14}$$

Replace $I(0)$ and $I(1)$ by $I(b)$ when both $I(0)$ and $I(1)$ occur. Let s be a fan-out stem with branches B_1, B_2, ..., B_k. The observation reachability function of the fan-out stem is calculated as

$$RO(s) = RO(B_i) \tag{15}$$

where $|RO(B_i)|$ is the minimum value among $|RO(B_1)|$, $|RO(B_2)|$, ..., $|RO(B_k)|$. The observation reachability function for the input of an inverter is the same as its output.

Let $j \in \{0, 1\}$, l and l' be the same gate in the first and second time frame of the circuit. Detection of a transition fault l/i ($i \in \{f, r\}$) sets the values j and $\overline{j}$ to l and l', respectively. The detectability of the transition fault l/i can be defined as the subset of inputs (including PIs in the second frame, PIs and PPIs in the first frame) used to detect the fault. The detectability measure of the transition fault l/i can be as

$$\det(l/i) = RC_j(l) \cup RC_{\overline{j}}(l') \cup RO(l') \tag{16}$$

where l and l' are the locations of the transition fault in the first and the second frame, as shown in Fig. 1. In (16), $j = 0$ and $j = 1$ represent the rising and falling transition faults, respectively.

The proposed input influence measure should be more accurate than the traditional SCOAP measure [2], according to which i-controllability and observability measures of a node l represent the minimum number of nodes must be specified to control a value on the node and propagate the value at l to a PO, respectively. The influence input measure presents the

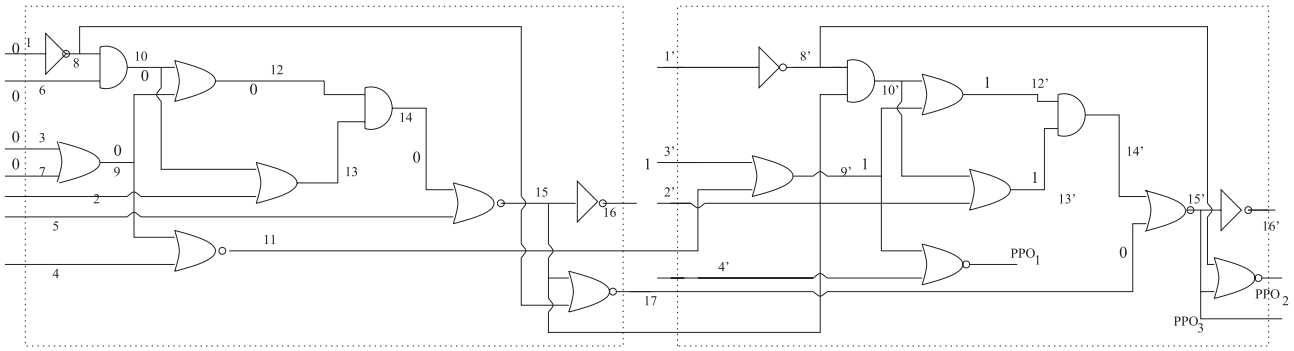

Fig. 3. Two-frame example circuit.

number of PPIs or PIs that must be specified to assign a value to a node, or propagate the value on a node to a PPO or PO. The proposed measure reflects the influence of reconvergent fan-outs, however, the SCOAP cannot measure.

Let us consider the circuit as shown in Fig. 2, whose two-frame circuit is shown in Fig. 3. We estimate the proposed measures for the two-frame circuit. The control reachability functions for all nodes in the circuit can be calculated as follows:

$$RC_0(9) = \{3, 7\},\ RC_1(9) = \{3\},\ RC_0(10) = \{\overline{6}\},$$
$$RC_1(10) = \{\overline{1}, 6\};\ RC_0(11) = \{4\},\ RC_1(11) = \{\overline{3}, \overline{4}, \overline{9}\};$$
$$RC_0(12) = \{\overline{3}, \overline{6}, \overline{7}\},\ RC_1(12) = \{3\};\ RC_0(13) = \{\overline{2}, \overline{6}\},$$
$$RC_1(13) = \{2\};\ RC_0(14) = \{\overline{2}, \overline{6}\},\ RC_1(14) = \{2, 3\};$$
$$RC_0(15) = \{5\},\ RC_1(15) = \{\overline{2}, \overline{5}, \overline{6}\};$$
$$RC_0(17) = \{\overline{1}\},\ RC_1(17) = \{1, 5\}.$$

As shown in the calculated control reachability functions, we use i to represent that the input i need to assign value 1, and $\overline{i}$ to value 0. The control reachability function for each node in the second frame can be estimated as follows:

$$RC_0(9') = \{4, \overline{3'}\},\ RC_1(9') = \{3'\};\ RC_0(10') = \{\overline{5}\},$$
$$RC_1(10') = \{1', \overline{2}, \overline{5}, \overline{6}\};\ RC_0(PPO_1) = \{4'\},$$
$$RC_1(PPO_1) = \{\overline{3'}, \overline{4'}, \overline{4}\};$$
$$RC_0(12') = RC_0(9') \cup RC_0(10') = \{4, \overline{3'}, \overline{5}\},\ RC_1(12') = \{3'\};$$
$$RC_0(13') = \{\overline{5}, \overline{2'}\},\ RC_1(13') = \{2'\};$$
$$RC_0(14') = \{4, \overline{2'}, \overline{3'}, \overline{5}\},\ RC_1(14') = \{2', 3'\};$$
$$RC_0(15') = \{2', 3'\},\ RC_1(15') = \{1, 5\};\ RC_0(PPO_2) = \{\overline{1'}\},$$
$$RC_1(PPO_2) = RC_0(15') \cup RC_0(8') = \{1', 2', 3'\}.$$

We shall not calculate the observation reachability functions for all nodes in the circuit. Let us consider the observation reachability measure $RO()$ for the node $13'$. We have

$$\begin{aligned} RO(13') &= RC_1(12') \cup RC_0(17) \\ &= \{3'\} \cup \{\overline{1}\} \\ &= \{\overline{1}, 3'\}. \end{aligned}$$

That is, input 1 in the first frame must be assigned value 0, and input $3'$ in the second frame must be assigned value 1 for the easiest way to observe the value of node $13'$. As for the detectability measure for the rising transition at node 12 as shown in Fig. 3

$$\begin{aligned} \det(12/r) &= RC_0(12) \cup RC_1(12') \cup RO(12') \\ &= RC_0(12) \cup RC_1(12') \cup RC_1(13') \cup RC_0(17) \\ &= \{\overline{1}, \overline{3}, \overline{6}, \overline{7}, 2', 3'\}. \end{aligned}$$

That is, to detect the rising transition fault at node 12, inputs $2'$ and $3'$ in the second frame must be assigned value 1, and 1, 3, 5, and 7 in the first frame must be specified value 0 according to the proposed input influence measure. The assignment on inputs exactly detects the rising transition at node 12.

III. Conflict-Avoidance-Driven ATPG for Transition Faults

We propose a conflict-avoidance ATPG scheme for LOC transition faults based on the input influence measure presented in Section II. Fig. 1 shows the two time frame circuit model for the rising transition fault l/r at node l. The stuck-at-1 fault is activated in the first frame. The fault $l/0$ is activated and its fault effect is propagated to either a PO or PPO in the second time frame.

As shown in Fig. 1, activation of the fault $l/1$ in the first time frame can specify lines in the area R_1. Activation of the fault $l/0$ and propagation of the activated fault effect in the second time frame can specify lines in the area R_3. It is necessary to specify some PPIs when activating the fault $l/0$ and propagating the activated fault effect in the second frame. The values of the PPIs must be justified to the PPIs and the PIs in the first frame.

We would like to introduce a new test pattern generation scheme based on conflict-avoidance. First, activation of the fault $l/1$ is justified to the PPIs and PIs in the first time frame. The new testability measure is used to guide backtracing. Let l/r be the target fault, which is located at the output of a 2-input AND gate. The node l must be assigned a value 0 to activate the $l/1$ fault in the first time frame. In order to satisfy the signal requirement $(l, 0)$, A or B can be assigned the value 0. Our method selects the input whose 0-control reachability function contains less PPIs. That is, input A is selected if $|RC_0(A)| < |RC_0(B)|$.

The PPO set PPO in the first time frame that can be influenced by the specified PPIs and PIs in the first frame. The subset PPI that needs to be avoided when justifying the signal requirements in the second time frame can be obtained

directly by replacing the output of each $ppo \in PPO$ by the data output PPI of the corresponding flip-flop.

Justification of all the necessary signal requirements in the second time frame needs to avoid the PPIs in PPI. This can be implemented when justifying the signal requirements to activate the fault in the second time frame as shown in Fig. 1.

Justifying the necessary signal requirements at the PPOs of the first time frame when generating a test vector for the $l'/0$ fault in the second time frame can produce some conflicts with the specified PPIs or PIs in the first frame. These conflicts can make the transition fault at l untestable. These conflicts include two different classes: 1) conflicts generated by justifying the signal requirements on the PPOs with the values when justifying the signal requirement for activation of the fault $l/1$ in the first time frame and 2) those conflicts produced when justifying the signal requirements on the PPOs in the first frame from the signal requirements to detect the fault $l'/0$ in the second frame.

No conflict can be produced between the signal requirement (ppo, i) on the PPO ppo and the signal requirement $(l, 0)$ on the fault site l in the first time frame if ppo and l do not have any common combinational predecessor. Let justification of the signal requirement $(l, 0)$ specify the pseudoprimary and PIs in IN. It is quite possible that a conflict can be prevented if there is no overlap between the i-control reachability function $RC_i(ppo)$ of a PPO ppo and IN. That is, $RC_i(ppo) \cap IN = \emptyset$.

No conflict exists between the signal requirement (ppo_1, i) on the PPO ppo_1 and the signal requirement (ppo_2, j) for $i, j \in \{0, 1\}$ in the first time frame if ppo_1 and ppo_2 do not have any common combinational predecessor. It is quite possible that there is no conflict if there is no overlap between the i-control reachability function on ppo_1 and the j-control reachability function on ppo_2. That is, $RC_i(ppo_1) \cap RC_j(ppo_2) = \emptyset$.

The subset of PPOs in the first time frame might need to be specified to generate a test for $l'/0$ in the second time frame for the target transition fault. As shown in Fig. 1, it is denoted as PPO. We introduce an IDG for the PPOs in PPO. The nodes of the IDG graph are all nodes in PPO. The nodes of the IDG are not all flip-flops in the circuit, but the specified PPIs in the second frame by generating the test of the stuck-at fault in the second frame. The nodes of the IDG are not all flip-flops in the circuit, but the specified PPIs in the second frame by generating the test of the stuck-at fault in the second frame. The number of nodes in the IDG should be much fewer than the number of FFs. Let the subset of PPOs PPO be specified by the necessary requirements to generate a test vector in the second time frame for the fault $l'/0$, shown in Fig. 1. We would like to introduce an IDG for the PPOs in PPO. The nodes of the IDG graph are all nodes in the PPO. Two nodes, ppo_1 and ppo_2 are connected if they have any common combinational predecessor in the circuit. All nodes in the IDG graph can be partitioned into separate subsets, where each subset of nodes is a connected component of the IDG graph. Two nodes that belong to two connected components in the IDG graph will not produce any conflicts when their corresponding signal requirements are justified.

Our method selects the connected component (ppo_1, ppo_2, ..., ppo_n), where there exist the signal requirements (ppo_1, i_1), (ppo_1, i_1), ..., (ppo_n, i_n) on ppo_1, ppo_2, ..., ppo_n, respectively. Here, $i_1, i_2, \ldots, i_n \in \{0, 1\}$. The signal requirements can have conflict with some of the PIs and PPIs in IN (the subset of inputs specified by activating the fault in the first frame). This information can be obtained by

$$\mathrm{OV} = |\mathrm{IN} \cap (RC_{i_1}(ppo_1) \cup \ldots \cup RC_{i_n}(ppo_n))|. \quad (17)$$

The connected components in IDG can be obtained easily by a single traversal of IDG. Our method first justifies the connected component that has more metric OV as presented in (17). All signal requirements related to the connected component on the PPOs in the first time frame are justified to the PIs and PPIs. This justification tries to avoid inputs in IN that are required conflict values that correspond to the specified inputs in IN. If no conflict with the previously specified values occurs, select the second connected component. The related signal requirements for the PPOs in the first time frame are justified again with the updated IN. Let justification of the signal requirements for the first connected component produce some conflict with the previously specified values. The ATPG algorithm typically regulates some proper assignments and backtracks. This process reiterates until all connected components in the IDG have been processed.

Justification of the signal requirements related to a connected component is determined as follows: try to backtrace the signal requirements away from the PIs or PPIs in the first time frame that has been specified up to now. This can be completed using the influence input measure. This process continues until all signal requirements at the PPOs of the first time frame circuit have been justified.

How to backtrace a signal requirement away from the specified inputs? Let IN be the set of PIs or PPIs that have been specified. Assign a value i on any one input of the gate g that can satisfy the signal requirement (g, j) of its output. Our method selects the input A whose i-control reachability function $RC_i(A)$ has less overlap with IN.

The ATPG algorithm goes back to the backtracking point when a conflict occurs in the same way as the base ATPG algorithm. The conflict-avoidance backtracing scheme is begun again. This process continues until a test has been generated or the fault is identified as untestable. The proposed method guarantees to avoid conflicts if possible because all possible trials can be proceeded before identifying a untestable fault.

Let us consider the example circuit as shown in Fig. 3 again. Let us generate the test pair for the rising transition fault at node 12 using the proposed test generation scheme. The node 12 must be set to 0 in the first circuit frame, which can be satisfied when setting the inputs 3, 6, and 7 to value 0. The stuck-at fault $12'/0$ must be detected in the second frame circuit. Therefore, value at $12'$ must be propagated to at least one PO or PPO. Therefore, $13'$ and 17 must be assigned 0. Node 17 can be set to 0 when setting one of 8 and 15 to value 1. The second choice can have conflict with the specified values in the first frame. We backtrace the signal requirement

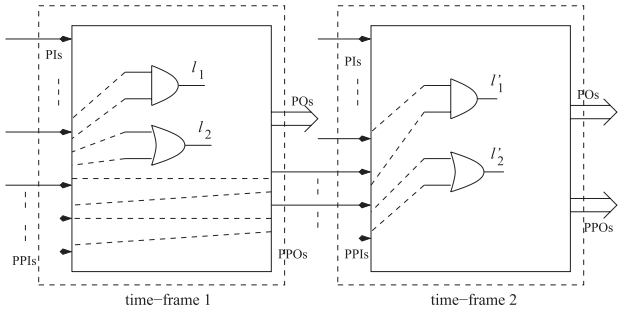

Fig. 4. Disjoint influence input subsets for two transition faults.

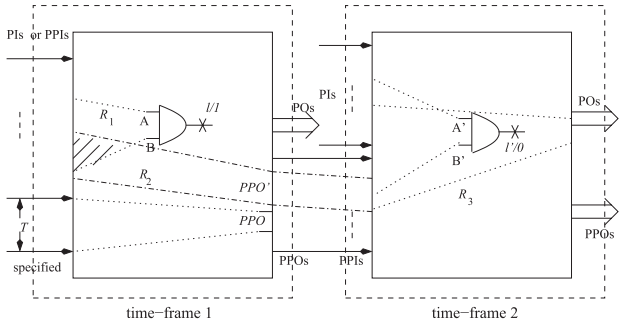

Fig. 5. Dynamic test compaction.

(17,1) to (1,0), which does not cause any conflict. We have

$$RC_1(12') \cup RC_0(13') \cup RC_0(17) = \{\overline{1}, 2', 3'\}$$
$$\{\overline{1}, 2', 3'\} \cap \{\overline{3}, \overline{6}, \overline{7}\} = \emptyset.$$

The proposed method can generate the test pair as, ({$\overline{1}$, $\overline{3}$, $\overline{6}$, $\overline{7}$}, {3′}) as shown in Fig. 3.

IV. Test Compaction for LOC Transition Fault Testing

We first introduce the dynamic test compaction scheme based on the influence input measure, and then a new static test compaction scheme is proposed.

A. Dynamic Test Compaction Based on the Influence Input Measure

Test compaction is to reduce the number of test vectors, which is classified into dynamic test compaction and static test compaction. Dynamic test compaction is completed during compact ATPG, which merges test vectors of two different faults when their test vectors are compatible. Static test compaction is to reduce the test count after ATPG has been completed. A test can be removed when the faults detected by the test can be detected by two or more other tests.

In general, two transition faults are called disjoint if they have disjoint influence inputs. That is, $\det(f_1) \cap \det(f_2) = \emptyset$ according to (16). In most cases, the tests for two transition faults can be compacted if the influence input subsets of two faults, calculated based on (16), are disjoint. Tests of two transition faults may still be compacted when the subsets of influence inputs for these two faults are conjoint, but all the conjoint inputs are assigned the same values to detect both faults.

Assume that the current test T specifies the PPIs or PIs in the first frame in T, and the influence input subset $\det(f)$ of fault f is calculated by (16). Initially, T is the test of the target fault. The fault f is called disjoint with T if $T \cap \det(f) = \emptyset$; otherwise, f is conjoint with test T, where $\det(f)$ represents the subset of PPIs or PIs required specifying to generate a test pair for f.

Fig. 4 shows the influence subsets of inputs for two transition faults that are disjoint. Test generation for one fault does not interfere with that for the other. Therefore, tests for any two faults can always be compacted into a single test if their structural influence input subsets are disjoint. Solely basing a decision about testing on the structural influence inputs is overly pessimistic. The detectability measure presented in (16) provides a more accurate estimation using the influence input measure, whose size is typically much smaller than that of the structural influence inputs.

ATPG test compaction requires a great deal of CPU time. To reduce the CPU time needed for test compaction, we can run the ATPG process once without any test compaction. All untestable transition faults can be removed from the fault list. It is not profitable for one to compact a redundant fault into a test repeatedly, which can waste CPU time. Our technique can be very profitable because test generation for all faults without dynamic test compaction requires a trivial amount of CPU time compared to that with dynamic test compaction according to our experimental results. More refined techniques can be used to remove the untestable transition faults for industrial designs if the no compaction ATPG process is still time consuming. For example, the proposed influence input measure can be used to identify untestable faults. A transition fault can be untestable if the detectability function as presented in (16) contains both i and $\overline{i}$ for at least one input. The compact ATPG process does not select the identified potential untestable faults identified by the influence input measure as the secondary target faults.

We propose a new dynamic test compaction scheme based on conflict avoidance and the influence input measure. The dynamic test compaction scheme is to generate a test for a new fault l/r under an initial test T. Let some of the PIs or PPIs in T be specified when compacting test of the new fault. Test compaction attempts to justify the value requirements of the fault in the first time frame and to enable all signal requirements to detect the fault in the second time frame, without causing conflicts. Techniques will be introduced to maximize the possibility of achieving this goal. Specifically, we propose a new dynamic test compaction scheme based on the influence input measure.

Let the current test vector be T, and ppo be the subset of PPOs in the first frame specified by T. We are considering compacting the test of the rising transition fault l/r into the current test T. As shown in Fig. 5, activation of the fault $l/1$ in the first time frame can specify the area R_1. Activation of the fault $l'/0$ and the propagation of the activated fault effect in the second time frame specify the area R_3 in the second time frame. In addition, it may be necessary to specify values at some PPIs in the second time frame that can be used to activate

the fault $l/0$ and to propagate the activated fault effect. The backtracing process tries to avoid the specified inputs PPO by the current test T. The required values at the PPIs of the second time frame by justifying the signal requirements in the second frame must be backtraced all the way to the PPIs and the PIs in the first time frame. Justifying the required values at the PPIs in the second time frame requires specifying values to those signals in the area R_2, as shown in Fig. 5. Areas R_2 and R_1 may have some overlap, which could cause conflicts and an untestable fault. Justifying the value requirements at the PPIs of the second time frame might also cause conflicts with those inputs used for activating the fault in the first time frame.

When justifying the signal requirement $(l, 0)$ in the first time frame as shown in Fig. 5, the backtrace avoids those inputs that have already been specified by the current test T. Assume that the required values in the second frame backtrace to the PPOs in PPO′. The IDG is established like the way in Section III. Just like the conflict-avoidance-based ATPG process, nodes of the IDG can form connected components. Our method selects the connected component whose detectability function has the maximum amount of overlap with T. All signal requirements related to the connected component on the PPOs in the first time frame are justified to the PIs and PPIs in the first time frame.

Clearly, value requirement related to a connected component of the IDG cannot cause any conflicts with any other connected component. The value requirements related to a connected component can still have too much constraints on the specified inputs up to now, which may have difficulty in avoiding the current test T. Signal requirements related to a single connected component in the IDG graph can be further partitioned. How to partition a connected component in IDG into multiple subsets is quite flexible, which should be helpful to avoid conflicts. It is better for each subset (ppo_1, ppo_2, …, ppo_n) to satisfy the following conditions: 1) ppo_1, ppo_2, …, ppo_n are connected in IDG; 2) $OV = \emptyset$ if possible; or 3) OV is as minimal as possible. Let ppo_1, ppo_2, …, ppo_k be assigned $v_1, v_2, \ldots, v_k$, and ppo_1, ppo_2, …, ppo_k be PPOs in the first frame. When multiple connected components have the same amount of overlap $|OV|$ with the current test T, our method selects the connected component whose signal requirements contain the greatest number of inputs that must be assigned values different from T. The overlap OV of a connected component in the IDG graph with T can be estimated based on (17), where IN is replaced by T.

Justification of the signal requirements for a selected connected component avoids assigning any values to the PIs or PPIs that have been specified by the current test T. When justifying the signal requirement (g, j) at the output of gate g, which needs the required value i at one of its inputs, the method selects the input A whose i-control reachability function $RC_i(A)$ has less overlap with T. The secondary target fault candidate set can be updated one or more times in the process of test compaction. Its optimal size can be determined by considering a tradeoff between the CPU time and compactness.

If no conflict with the previously specified values occurs after the justification of the first selected connected component, select the next connected component whose signal requirements have the most overlap $|OV|$ with the updated test T based on the detectability function for justification. If a conflict occurs in the justification process, the ATPG algorithm insures that the proper assignments and backtracking are used. This process continues until all connected components of the IDG have been processed and the test of the target transition fault's test has been successfully compacted. Otherwise, we conclude that the transition fault test cannot be compacted.

In either case, the method selects another fault in ST and continues the process until all the faults in ST have been tried. After one compacted test is generated, the method selects another primary target fault and the secondary target fault set ST from the remaining, uncovered faults in the same manner until compacted tests for all faults have been generated for all faults.

Fig. 6 shows the details of the new dynamic test compaction scheme. The new test compaction scheme is based on the estimated influence input function. First, a transition fault f with a minimum influence input subset based on (16) is selected as the primary target fault, as presented in Step 1. Generate a test pair T for the selected target fault. All faults

dynamic-test-compaction()

1) While the remaining fault set is not empty, do steps (2)-(7).
2) Select a transition fault f with the minimum number of influence inputs $|det(f)|$ based on Equation (16) as the primary target fault. Generate a test pair T for the transition fault f.
3) $T' \leftarrow T$. Select secondary target fault candidates from the remaining fault set, whose influence inputs has the minimum overlap with T'. T' is updated by merging the influence inputs of the selected secondary target fault with T. Continue this process until the given number of secondary target faults have been selected and put in ST.
4) If there exists $f' \in ST$ with $T \cap IC(f') = \emptyset$. Generate the test pair of f' under the constraints of T, update T. Select the one with the minimum $|det(f')|$ when there is a tie for this. Continue the above process until no such fault is available.
5) If no f' exists in ST with $T \cap IC(f') = \emptyset$. Select the fault $f' \in ST$ with $T \cap det(f') = \emptyset$. Select the one with the minimum $|det(f')|$ when there is a tie for this. Continue the above process until no such fault is available.
6) Select a fault $f' \in ST$ which has the minimum conjoint with the current test T, and all conjoint inputs are required compatible values with T. Try to compact its test into the current test T. Select the one that may specify the minimum number of new inputs when there is a tie for this. Continue this process until no such fault is available.
7) Select the fault $f' \in ST$ which has the minimum conjoint with the current test T, which are required incompatible value with T. Try to compact its test into the current test T. Select the one whose test may specify the minimum number of new inputs when multiple choices exist. Continue the above process until no such fault is available.
8) Keep the final compact test pair T in the test set. Do fault simulation on the remaining fault list if it is still not empty.

Fig. 6. Procedure of the new test compaction scheme.

whose influence input functions have a minimum amount of overlap with T are placed into the secondary target fault set ST, as given in Step 3. The size of the secondary target fault set is determined based on the tradeoff between the compactness and CPU time. Fig. 6 shows the procedure for the new test compaction.

Let $IC(f)$ and $\det(f)$ be the subsets of influence inputs of the transition fault f based on structural analysis and the detectability measure as presented in (16). We use these heuristics for dynamic test compaction. As stated earlier, a test T (the specified input set) for the primary target fault f is generated based on the detectability measure, which minimizes the number of PIs and PPIs with a value assignment. The scheme to select the secondary target fault set ST is a preprocessing of the test compaction process, tests of which are easy to compact into the current test. If there exists a fault $f' \in ST$ whose influence inputs based on structural analysis is disjoint with the current test T, f' is selected as the next secondary target and its test is generated and compacted into the current test T. The technique to select the secondary target fault set and faults to compact into the current test pair is completed based on the influence input measure presented in Section II, which is quite different from the dynamic test compaction scheme for path delay faults in [33]. The dynamic test compaction scheme for path delay faults was proposed based on output cones.

If no such fault exists while a fault f' still exists in ST, $T \cap \det(f') = \emptyset$, where $\det(f')$ is calculated by the influence input measure as presented in (16). Actually, $\det(f')$ is just a subset of PPIs or PIs with specified values. Then, f' is selected as the next target and its test is generated and compacted into the current test T as stated in Step (5).

We should still attempt to compact the test of a fault into the current test even when there exists no fault $f' \in ST$ whose subset of influence inputs $\det(f')$ is disjoint with the current test as stated in the above two cases. Our method selects a secondary target fault in the following way. A fault $f' \in ST$ that has the minimum conjoint with the current test is selected, where all conjoint inputs according to the influence measure are assigned the compatible values compared with the current test T. That is, for any conjoint input IN, $in \in \det(f') \cup RO(l')) \cap T$, it is necessary to assign IN the same value as the current test T. The test of the selected fault is tried to compact into the current test. Our method iteratively selects the next fault $f' \in ST$ based on the above three criteria, whose influence inputs are conjoint with T. Continue the above process until the compaction of all secondary target faults in ST has been attempted.

Our method selects a fault $f' \in ST$ that has the minimum conjoint with the current test, where all conjoint inputs according to the influence measure are assigned the incompatible values compared with the current test T. Note that the minimum conjoint as presented in step (7) represents the minimum number of inputs that are required assigning values different from those of T. That is, for any conjoint input IN, in $\in \det(f') \cup RO(l')) \cap T$, it is necessary to assign IN different values as the current test T. The test of the selected fault is tried to compact into the current test. Continue the above process until the compaction of all secondary target faults in ST has been attempted.

Let l/r be the selected as the secondary target fault as shown in Fig. 1, which is located at the output of a 2-input AND gate. We assume that the current test T has already specified values at some PIs and PPIs. The node l' must be assigned value 1 to activate the $l'/0$ (l' stuck-at 0) fault in the second time frame. Our method selects the input whose 0-control reachability function contains fewer PPIs specified by the current T. That is, input A' is selected if $|RC_1(A') \cap T| < |RC_1(B') \cap T|$.

The signal requirements at the PPOs of the first time frame when compacting the test vector of the transition fault might produce some conflicts. These conflicts include four different classes: 1) those generated by signal requirements at the PPOs from the second frame that conflict with the signal requirement generated by activating the fault $l/1$ in the first time frame; 2) conflicts produced among the signal requirements at the PPOs driven from the second time frame; 3) those among the current test and the signal requirements on the PPOs driven from the second time frame that occur when detecting the fault $l'/0$ (l' stuck-at 0) in the second frame at the same line of l; and 4) conflicts between the signal requirements for activating the fault in the first time frame and the specified inputs corresponding to the current test. The conflicts resulting from 1) and 2) make the fault untestable. The conflicts resulting from 3) and 4) make it impossible to compact the test of the target fault into the current test.

Justification of the signal requirements for a selected connected component avoids assigning any values to the PIs or PPIs that have been specified by the current test T. When justifying the signal requirement (g, j) at the output of gate g, which needs the required value i at one of its inputs, the method selects the input A whose i-control reachability function $RC_i(A)$ has less overlap with T. The secondary target fault candidate set can be updated one or more times in the process of test compaction. Its optimal size can be determined by considering a tradeoff between the CPU time and compactness.

Fig. 7 shows the example for dynamic test compaction still in the same circuit as shown in Fig. 3. Let us consider the rising transition faults at 11 and 14. As shown in Fig. 7, the test pair for the rising transition fault at node 11 is generated first, which is $(\{4\}, \{\overline{3'}, \overline{4'}\})$. The rising transition fault at 14 must generate a value 0 at 14, which does not have any conflict with the specified inputs because we have $RC_0(14) = \{\overline{2}, \overline{6}\}$. To assign value 0 to node $12'$ in the second frame, nodes $10'$ must be set to 1, which can be met by assigning value 1 to $8'$ and 15. This can be satisfied by setting $1'$ to value 1 and 5 to value 0. In addition, $13'$ must be set to value 1 and 17 must be set 0. This can be met by setting $2'$ to 1 additionally. Finally, the compact test pair is $(\{\overline{2}, 4, \overline{5}, \overline{6}\}, \{\overline{1'}, 2', \overline{3'}, \overline{4'}\})$.

B. New Static Test Compaction Scheme

Dynamic test compaction is to generate a single test pair for multiple faults. Different transition faults, that must specify different subsets of PIs or PPIs to generate tests for them,

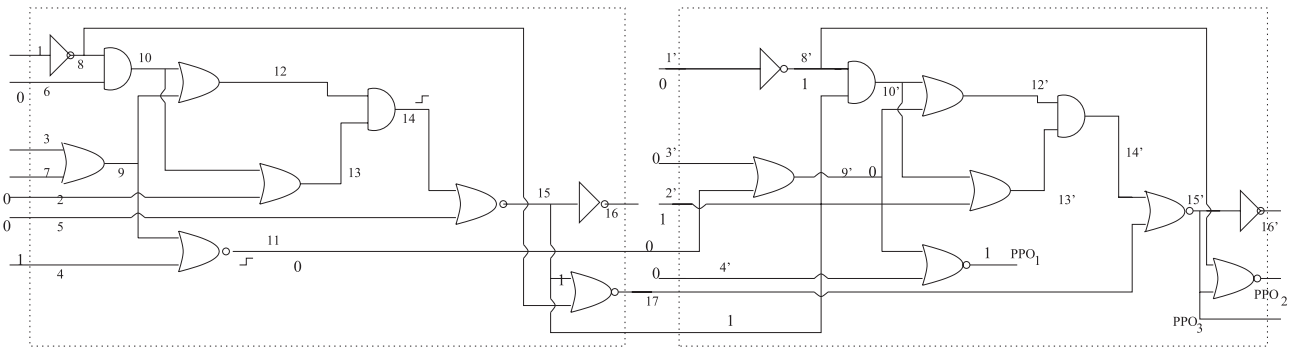

Fig. 7. Case study for the dynamic test compaction scheme.

static-test-compaction()
{
1) Do fault simulation without fault dropping on all tests T_{org} for the whole fault list.
2) Select the essential tests for the test set T_{org}, put all essential tests into the compact test set T_c; $T_{org} \leftarrow T_{org} - T_c$.
3) while $T_{org} \neq \emptyset$, do step (4)
4) $T_{org} \leftarrow T_{org} - \{t\}$, check whether all faults detected by t are detected by the tests in T_c. If not, $T_c \leftarrow T_c \cup \{t\}$.
5) $T_{org} \leftarrow T_c$; if the total number of tests removed is more than a threshould, go to step 2.
6) Return the final compacted test set T_{org}.
}

Fig. 8. New static test compaction scheme.

can be detected by the same test. The first selected target fault is the primary target fault, and all faults selected into the subset of faults F_1 as shown in Fig. 6 are secondary target faults. A dynamic test compaction scheme was proposed in Section III, which avoids the specified PIs or PPIs for the secondary target faults in the process of backtracing.

The test set can still be reduced after the compact test set has been generated, which is called static test compaction. We propose a new and simple static test compaction scheme. A test is an essential test if it covers at least one fault that is not covered by any other tests.

Our method first does fault simulation without fault dropping on all tests T_{org} for the whole fault list, which gets the whole detected fault list for all tests. Our method then selects the essential tests. All essential tests are put into the initial compacted test set T_c. For each test $t \in T_{\text{org}}$, our method checks whether all faults detected by t are detected by the tests in T_c. If so, remove t from T_{org}. Otherwise, put t into T_c. After all tests have been handled, our method sets T_{org} as the compacted test T_c.

Again, our method identifies the essential tests for T_{org} because some tests can be removed from the test set. Removal of some new tests from the test set can introduce some new essential tests, however, all previous essential tests are still essential. Put all new essential tests into T_c. Our method checks all tests again in another order in the same way. This process continues until the number of tests removed in the current round is no more than a threshold.

TABLE I
STATISTICS OF THE CIRCUITS

circuit	*faults*	FFs	PIs	POs	*no compaction*		
					FC	vec	CPU
s1423	2512	74	17	5	89.25	160	0.1
s5378	6988	179	35	49	90.13	479	1.0
s9234	11328	228	19	22	81.22	881	6.9
s9234.1	11328	211	36	39	83.16	906	5.2
s13207	15602	669	31	121	82.26	-	-
s13207.1	15602	638	62	152	88.95	1084	8.6
s15850	19046	597	14	87	74.34	710	16.9
s15850.1	19046	534	77	150	85.35	943	21.7
s35932	62798	1728	35	320	84.26	138	111.9
s38417	49738	1636	28	106	96.92	1992	179.9
s38584	61254	1452	12	278	90.34	1850	224.6
s38584.1	61254	1426	38	304	92.53	1970	175.2
b17	115110	1415	37	97	74.04	1891	1527
b18	327944	3320	37	23	77.12	4763	25281
b19	499924	6642	24	30	84.59	-	-
b20	80604	490	32	22	86.64	2239	361
b22	119806	735	32	22	88.04	2897	879
systemcaes	39556	190	1128	1147	94.25	697	25.6
tv80	29116	359	60	140	87.08	1424	84.7
aes_core	70210	530	672	792	99.84	1576	71.1
wb_cinmax	210540	770	1129	1416	99.19	7037	2602
mem_ctrl	42734	1083	440	1231	98.30	1870	44.4
usb_funct	66918	1746	104	19	98.01	2951	169.6
dma	106440	2192	1626	3396	95.83	4378	4058
ac97_ctrl	82462	2199	253	2216	93.48	1590	128
pci_bridge32	139968	3359	160	207	97.62	3022	440
des_perf	448789	8746	233	64	99.998	638	471
ethernet	480764	10554	94	115	99.47	13715	5074.2
vga_lcd	785428	17079	87	109	99.64	16555	9302.2

The proposed static test compaction scheme as shown in Fig. 8 first puts all essential tests into the compact test set T_c. For each test $t \in T_{\text{org}}$, t is removed if each of the faults detected by t can also be covered by at least one test in T_c. Otherwise, t is put into T_c. The essential tests generated early are still essential after some tests are removed from the test set. However, some new essential tests can be produced. After each round of test compaction, our method identifies the new essential tests because some tests have been removed. In any case, the essential tests in the previous round of test compaction are still essential tests.

V. EXPERIMENTAL RESULTS

The proposed conflict-avoidance-driven dynamic test compaction schemes for LOC transition fault testing have been implemented using a Dell Precision 490 workstation. Table I presents the test generation results for the proposed method without any test compaction technique, and the statistics of the benchmark circuits. As shown in Table I, columns faults,

TABLE II
PERFORMANCE COMPARISON ON FAULT COVERAGE AND TEST SET SIZE WITH PREVIOUS METHODS

Circuit	*proposed*		[15]		[32]		[9]		[23]	
	vec	FC	vec	FC	vec	FC	vec	FC	vec	FC
s1423	**32**	**90.21**	-	-	135	86.1	92	89.25	59	88.55
s5378	**90**	90.13	-	-	248	78.8	236	91.08	278	**91.87**
s9234	**213**	81.22	528	**86.20**	350	81.5	-	-	428	80.75
s9234.1	**166**	**90.34**	692	79.11	444	82.6	354	86.48	-	-
s13207	**175**	**82.26**	253	78.69	356	79.8	-	-	-	-
s13207.1	**96**	**88.95**	-	-	-	-	603	87.52	-	-
s15850	**109**	**74.34**	365	70.45	220	69.9	-	-	379	70.77
s15850.1	**160**	**88.95**	-	-	-	-	304	84.17	-	-
s35932	32	84.26	**30**	81.96	72	84.7	64	85.98	**7**	**87.21**
s38417	**162**	96.92	737	97.27	227	98.0	286	97.97	-	-
s38584	**367**	**90.34**	692	79.11	444	82.6	-	-	867	87.71
s38584.1	**386**	**92.53**	-	-	-	-	416	91.99	-	-

FFs, PIs, and POs represent the numbers of transition faults, flip-flops, PIs, and POs, respectively. The number of faults is the same as presented in [6] and [9] for almost all circuits. Columns no compaction presents fault coverage, the number of test vectors, and CPU time (seconds) to generate the test sets without any test compaction.

Table II presents comparison of the test compaction method with previous methods [9], [15], [23], and [32]. The earlier compact test generation method in [9] provides good enough results compared with other early works. The method in [15] includes IR-drop tolerance, and the method in [32] considered low capture power. However, both method still provides very compact test sets, which is the most important reason why we still compare the proposed method to the ones in [15] and [32]. Pomeranz in [23] proposed a compact test generator for multiple capture cycle broadside transition tests, which reduces the number of tests. The results indicate that the proposed test compaction scheme outperforms the previous methods for almost all circuits except s35932 on test set size compared with the method in [15]. The proposed dynamic test compaction scheme obtains fewer test vectors than the method in [9] for all circuits. Compared with the method in [23], our method works better on the pattern count vec for all circuits except s35932. However, the method in [23] obtains much better pattern count for circuit s35932. The methods in [24] and [25] present compact mixed test sets with both LOC and skewed-load tests, which cannot be used to compare with the proposed method.

Results for some ISCAS89 circuits, the largest ITC99 circuits and some IWLS2005 circuits are presented in Table III. Our method is compared with two commercial broadside transition fault ATPG tools on the pattern count (vec) and fault coverage. The commercial ATPG tools run test compaction in multiple rounds, which does not produce the whole CPU time. This is the reason why the CPU time for the commercial ATPG tools is not provided. As shown in Table III, vec_1 and vec present the pattern count for the dynamic test compaction scheme and the pattern count after the static test compaction scheme has been combined. In comparison with the commercial transition ATPG tools, the proposed method obtains much lower pattern count for almost all circuits. The proposed method obtains compatible fault coverage compared with the commercial ATPG tool com2 for the ISCAS89 circuits. However, the proposed method produces apparently smaller

TABLE III
PERFORMANCE EVALUATION OF THE PROPOSED COMPACT ATPG METHOD

circuit	*proposed*				*Com1*		*Com2*	
	FC	vec_1	vec	CPU(s)	FC	vec	FC	vec
s1423	90.21	64	32	2.3	89.01	96	87.44	103
s5378	90.13	145	128	15.4	89.04	271	90.22	198
s9234	81.22	326	213	195	87.78	252	84.21	384
s9234.1	83.16	316	187	88.5	86.44	175	90.26	369
s13207	82.26	546	64	262	92.78	569	81.33	362
s15850	74.34	261	109	275	98.43	133	72.54	208
s35932	84.26	53	32	175	98.12	80	86.47	72
s38417	96.92	191	162	7111	98.04	1192	98.46	229
s38584	90.34	558	367	3937	97.5	622	90.26	369
b17	74.04	764	574	2167	91.57	1421	88.07	1180
b18	77.12	1421	1221	42057	81.72	6330	84.1	1348
b19	84.59	11577	9652	134719	86.90	12047	86.72	12693
b20	86.64	1188	851	5942	88.68	1063	85.23	1181
b22	88.04	1471	1148	8067	87.19	1632	86.48	1589
systemcaes	94.25	410	288	509	93.09	354	89.31	392
tv80	87.08	944	352	2002	90.14	1076	89.53	1101
aes_core	99.84	854	453	506	92.01	837	97.74	722
wb_cinmax	99.19	2036	1128	55009	98.48	1784	96.76	1855
mem_ctrl	98.30	787	342	1029	27.65	117	89.46	722
usb_funct	98.01	543	352	505.9	95.2	1859	89.93	531
dma	95.83	1512	1088	15252	99.92	2398	88.72	1460
ac97_ctrl	93.48	283	204	1009	92.53	568	80.74	252
pci_bridge32	97.62	845	630	6173	96.62	1660	84.49	634
des_perf	99.998	213	192	9325	100	606	96.25	351
ethernet	99.47	4752	3501	95178	99.89	6629	58.95	3557
vga_lcd	99.64	12543	9045	174638	99.11	10147	88.52	5645

TABLE IV
PERFORMANCE EVALUATION FOR LOC TRANSITION TEST GENERATION WITH CONSTANT PRIMARY INPUTS

Circuit	*proposed*						FC[26]	
	faults	FC	vec_1	vec	CPU	arbit	const	arbit
s13207	13042	68.76	469	72	213	82.26	79.95	81.53
s35932	60334	80.95	48	30	247	84.26	71.79	87.21
s38417	48042	93.61	253	142	6340	96.92	95.78	97.05
s38584	58174	85.79	450	326	3170	90.34	69.06	88.20

test sets for almost all circuits, and better fault coverage for most IWLS2005 circuits. In addition, the proposed method does not have any aborted fault for all circuits.

As shown in Table III, the results indicate that the proposed test compaction scheme significantly reduces the number of test vectors without any fault coverage loss. A combination of the dynamic and static test compaction schemes can generate a very compact test set in reasonable CPU time. The proposed ATPG method detects all testable faults for all circuits very quickly when no test compaction technique is included. It is believed that the input influence measure-based dynamic test compaction scheme and the new static test compaction scheme are very effective.

Some LOC transition ATPG tools disallow the PIs change for both vectors of a test pair [26]. Table IV presents the fault coverage of the test generator with constant PIs. The work in [26] does not present the number of test vectors. The fault coverage, the test set size before static test compaction (vec_1), the test set size (vec) after static test compaction has been combined, and CPU (seconds) of the proposed compact test generator are presented when the PIs for both circuit frames hold constant.

VI. CONCLUSION

We propose a new compact test generation method for LOC delay testing of transition faults based on a new influence

input measure. This test generation method is based on the idea of conflict avoidance. The influence input measure, the subset of inputs required specifying to detect a transition fault in LOC scan testing, was provided to guide dynamic test compaction. Potential conflicts produced by the value requirements at the PPOs are avoided through the use of an IDG. The new dynamic test compaction scheme compacts tests for as many faults as possible into the current test. The experimental results indicate that the proposed method achieves better results than previous methods.

REFERENCES

[1] N. Ahmed and M. Tehranipoor, "Improving transition delay test using a hybrid method," *IEEE Design Test Comput.*, vol. 23, no. 5, pp. 402–412, May 2006.

[2] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing*. Boston, MA, USA: Kluwer, 2000.

[3] Z. Chen and D. Xiang, "A novel test application scheme for high transition fault coverage and low test cost," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 6, pp. 966–976, Jun. 2010.

[4] Z. Chen, K. Chakrabarty, and D. Xiang, "MVP: Minimum-violations partitioning for reducing capture power in at-speed delay-fault testing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1762–1767, Nov. 2011.

[5] Z. Chen and D. Xiang, "A novel test application scheme for high transition fault coverage," in *Proc. 27th IEEE VLSI Test Symp.*, May 2009, pp. 146–151.

[6] K. T. Cheng, "Transition fault testing for sequential circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 12, pp. 1971–1983, Dec. 1993.

[7] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. 32, no. 12, pp. 1137–1144, Dec. 1983.

[8] P. Goel, "An implicit enumeration algorithm to generate tests for combinational circuits," *IEEE Trans. Comput.*, vol. 30, no. 3, pp. 215–222, Mar. 1981.

[9] I. Hamzaoglu and J. H. Patel, "Compact two-pattern test set generation for combinational and full scan circuits," in *Proc. IEEE Int. Test Conf.*, Oct. 1998, pp. 944–953.

[10] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 8, pp. 957–963, Aug. 2000.

[11] Y. Han, Y. Hu, X. Li, and H. Li, "Embedded test decompressor to reduce the required channels and vector memory of tester for complex processor circuit," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 5, pp. 531–540, May 2007.

[12] S. Jin, Y. H. Han, H. Li, and X. Li, "Unified capture scheme for small delay defect detection and aging prediction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 5, pp. 821–833, May 2013.

[13] A. Krstic, J. Liou, K. T. Cheng, and L. C. Wang, "On structural vs. functional testing for delay faults," in *Proc. IEEE 4th Int. Symp. Qual. Electron. Design*, Mar. 2003, pp. 438–441.

[14] W. Kunz and D. K. Pradhan, "Accelerated dynamic learning for test pattern generation in combinational circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 5, pp. 684–694, May 1993.

[15] J. Lee, S. Narayan, M. Kapralos, and M. Tehranipoor, "Layout-aware, IR-drop tolerant transition fault pattern generation," in *Proc. ACM/IEEE Design Autom. Test Eur.*, Mar. 2008, pp. 1172–1177.

[16] Y. C. Lin, F. Lu, and K. T. Cheng, "Pseudofunctional testing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 8, pp. 1535–1546, Aug. 2006.

[17] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderkneck, and N. Tamarapalli, "High-frequency, at-speed scan testing," *IEEE Design Test Comput.*, vol. 20, no. 5, pp. 17–25, Sep./Oct. 2003.

[18] J. J. Liou, L. C. Wang, K. T. Cheng, J. Dworak, M. R. Mercer, R. Kapur, and T. W. Williams, "Enhancing test efficiency for delay fault testing using multiple-clocked schemes," in *Proc. ACM/IEEE 39th Design Autom. Conf.*, Jun. 2002, pp. 371–374.

[19] J. J. Liou, A. Krstic, Y. M. Jiang, and K. T. Cheng, "Modeling, testing and analysis for delay defects and noise effects in deep submicron devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 6, pp. 756–769, Nov. 2003.

[20] X. Liu, M. Hsiao, S. Chakravarty, and P. J. Thadikaren, "Efficient techniques for transition fault testing," *ACM Trans. Design Autom. Electron. Syst.*, vol. 10, no. 2, pp. 258–278, Apr. 2005.

[21] W. Mao and M. D. Ciletti, "Reducing correlation to improve coverage of delay faults in scan-path design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 5, pp. 638–646, May 1994.

[22] T. Niermann and J. H. Patel, "HITEC: A test generation package for sequential circuits," in *Proc. EDAC*, Feb. 1991, pp. 214–218.

[23] I. Pomeranz, "Generation of multi-cycle broadside tests," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1253–1257, Aug. 2011.

[24] I. Pomeranz, "Generation of mixed test sets for transition faults," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 10, pp. 1895–1899, Oct. 2012.

[25] I. Pomeranz, "On test compaction of broadside and skewed-load test cubes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 9, pp. 1705–1714, Sep. 2013.

[26] I. Pomeranz, "Undetectable transition faults under broadside tests with constant primary input vectors," *IET Comput. Digit. Tech.*, vol. 6, no. 2, pp. 78–85, Mar. 2012.

[27] J. Savir, "Broad-side delay test," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 3, pp. 368–372, Sep. 1994.

[28] M. H. Schulz, E. Trischler, and T. M. Sarfert, "SOCRATES: A highly efficient automatic test pattern generation system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 7, no. 1, pp. 126–137, Jan. 1988.

[29] Y. Shao, I. Pomeranz, and S. M. Reddy, "On generating high quality tests for transition faults," in *Proc. IEEE 11th Asian Test Symp.*, Nov. 2002, pp. 1–8.

[30] G. L. Smith, "Model for delay faults based upon paths," in *Proc. IEEE Int. Test Conf.*, Jun. 1984, pp. 342–349.

[31] Z. Wang and D. Walker, "Dynamic compaction for high quality delay test," in *Proc. 26th IEEE VLSI Test Symp.*, May 2008, pp. 243–248.

[32] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L. T. Wang, K. K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing," in *Proc. IEEE ITC*, Nov. 2005, pp. 1–10.

[33] D. Xiang, K. Li, H. Fujiwara, and J. Sun, "Generating compact robust and non-robust tests for complete coverage of path delay faults based on stuck-at tests," in *Proc. 24th IEEE ICCD*, Oct. 2006, pp. 446–451.

[34] 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. Design Autom. Electron. Syst.*, vol. 17, no. 2, pp. 1–18, Apr. 2012.

[35] D. Xiang, J. Li, K. Chakrabarty, and X. Lin, "Test compaction for small delay defects using an effective path selection scheme," *ACM Trans. Design Autom. Electron. Syst.*, vol. 18, no. 3, pp. 1–44, Jul. 2013.

[36] B. Yin, D. Xiang, and Z. Chen, "New techniques for accelerating small delay defect ATPG and test compaction," in *Proc. 22th Int. Conf. VLSI Design*, Jan. 2009, pp. 221–226.

[37] Q. Xu and N. Nicolici, "DFT infrastructure for broadside two-pattern test of core-based SoCs," *IEEE Trans. Comput.*, vol. 55, no. 4, pp. 470–485, Apr. 2006.

Dong Xiang (SM'13) received the B.S. and the M.S. degrees in computer science from Chongqing University, Chongqing, China, in 1987 and 1990, respectively. He received the Ph.D. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1993.

He was a Post-Doctoral Researcher with Concordia University, Montreal, QC, Canada from 1994 to 1995, and the University of Illinois, Urbana-Champaign, IL, USA, from 1995 to 1996. He was a JSPS Invitation Fellow with the Nara Institute of Science and Technology, Ikoma, Japan, from April to September 2003. He was an Associate Professor with the Institute of Microelectronics from October 1996 to March 2003. He is currently a Professor with the School of Software, Tsinghua University, Beijing, China. His current research interests include design and test of digital systems, fault-tolerant computing, distributed computing, and computer networking. He authored *Digital Systems Testing and Design for Testability* in 1997.

Dr. Xiang received the National Outstanding Young Scientist Award from the National Science Foundation of China in 2004. He is a Senior Member of the Chinese Computer Federation and a member of ACM.

Wenjie Sui received the bachelor's degree in engineering from Chongqing University, Chongqing, China, in 2011.

He is currently pursuing the master's degree in software engineering with the School of Software, Tsinghua University, Beijing, China. His current research interests include delay testing and design for testability.

Boxue Yin received the bachelor's degree in computer engineering from Sijiazhuang Institute of Technology, Hebei, China, in 2007, and the master's degree in software engineering from Tsinghua University, Beijing, China, in 2010.

His current research interests include delay testing and design for testability.

Kwang-Ting Cheng (S'88–M'88–SM'98–F'00) received the B.S. degree in electrical engineering from National Taiwan University, Tainan, Taiwan, in 1983, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 1988.

He was with Bell Laboratories, Murray Hill, NJ, USA, from 1988 to 1993. He was a faculty member with the University of California, Santa Barbara (UCSB), CA, USA, in 1993, where he is currently a Professor. He was the Founding Director of UCSB's Computer Engineering Program from 1999 to 2002 and the Chair of the Electrical Communication Engineering Department from 2005 to 2008. His current research interests include design verification, test, resiliency, postsilicon debug, and multimedia computing.

Dr. Cheng was a recipient of the Best Paper Awards with the AT&T Conference on Electronic Testing in 1987, the Design Automation Conference in 1994 and 1999, the Conference of Design Automation and Test in Europe in 2003, the IEEE Asian Test Symposium in 2007, the International Symposium for Flexible Electronics and Displays in 2007, and the IEEE VLSI Test Symposium in 2008. He received the Annual Best Paper Award in the Journal of Information Science and Engineering in 2001 and the UCSB College of Engineering Outstanding Teaching Award from 2004 to 2005. He served as an Editor-in-Chief of the IEEE DESIGN AND TEST OF COMPUTERS from 2006 to 2009, and served as an Associate Editor of the ACM TRANSACTIONS ON DESIGN AUTOMATION OF ELECTRONIC SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, Formal Methods in System Design, the Journal of Electronic Testing: Theory and Applications, and Foundations and Trends in Electronic Design Automation.