J Electron Test (2014) 30:763–780
DOI 10.1007/s10836-014-5490-4

Diagnostic Test Generation for Transition Delay Faults Using Stuck-At Fault Detection Tools

Yu Zhang · Bei Zhang · Vishwani D. Agrawal

Received: 10 July 2014 / Accepted: 21 October 2014 / Published online: 31 October 2014
© Springer Science+Business Media New York 2014

Abstract By adding a few logic gates and one or two modeling flip-flops to the circuit under test (CUT), we create a detection or diagnostic automatic test pattern generation (ATPG) model of transition delay faults usable by a conventional single stuck-at fault test pattern generator. Given a transition delay fault pair, the diagnostic ATPG model can either find an exclusive test or prove the equivalence of the fault pair. Our work offers advantages over existing work. First, the detection of a transition delay fault or the diagnosis of a fault pair can be modeled in only one instead of two or four time-frames of the CUT. Second, an exclusive test can be generated under either launch off capture (LOC) or launch off shift (LOS) mode for a full-scan sequential circuit. Third, the proposed ATPG models can be expanded into two time frames to facilitate the use of combinational ATPG tools, though with lower modeling complexity than was possible before. As a result, the percentage of distinguished transition delay fault pairs is larger and the proposed automatic exclusive test generation system is more time-efficient.

Keywords ATPG · Delay faults · Digital circuit testing · Fault diagnosis · Test generation · Timing defects · Transition faults

Responsible Editor: M. Tehranipoor

Y. Zhang · B. Zhang (✉) · V. D. Agrawal
Department of Electrical and Computer Engineering,
Auburn University, Auburn, AL 36849, USA
e-mail: bzz0004@tigermail.auburn.edu

Y. Zhang
e-mail: yzz0009@tigermail.auburn.edu

V. D. Agrawal
e-mail: vagrawal@eng.auburn.edu

1 Introduction

Due to increases in logic density, speed and the time to market pressure of modern VLSI chips, manufacturers must quickly bring up the yields. Identifying the locations of defects in failing devices in a systematic way can help find the problems within manufacturing process and thus help improve the yield. Due to innumerable nets within a VLSI chip, this task is complex and can take a long time. Fault diagnosis is a necessary step in isolating the most likely defective net locations so that defect identification and elimination can be done.

When a manufactured device is tested using automatic test equipment (ATE), it either passes all applied tests or fails some tests. In the latter case, the ATE data lists failing tests and the corresponding observation points where the device response differed from the expected value. These observation points can be either primary outputs (POs) or scan flip-flops. Based on the failing test and failing observation points, a diagnosis algorithm then finds a list of suspected faults that could have caused the reported failures.

Diagnosis algorithms are broadly classified into two categories: (1) Effect-cause algorithms [20, 48, 49]. These algorithms consist of four steps. First, list suspected faults by tracing fan-in cones from failing POs. Second, simulate the suspected faults. Third, rank the suspected faults based on how strongly a fault explains the failing responses. Lastly, compress the list of suspected faults by filtering out low ranking suspects. (2) Cause-effect algorithms [2, 11, 12, 15, 18, 19, 27, 34, 36, 37, 46, 47, 50–53, 55]. In these algorithms, the simulation data for all modeled faults and all tests is saved as a dictionary of fault syndromes (or signatures) to be matched with the failing device response. The technique proposed in this work aims at generating exclusive tests [2] to diagnose transition delay faults

(TDFs), with emphasis on improving the diagnostic metrics of the cause-effect dictionary. More details and relative work of cause-effect diagnosis algorithm are introduced in Section 2.

For a single-output circuit, an *exclusive test* for a pair of faults is defined as a test that detects one fault but not the other [2]. Its primary intent is to distinguish the fault pair. For a multiple output circuit, this definition applies separately to each output. An exclusive test for a fault pair can distinguish them as long as they are not being detected at the same set of outputs. Since the purpose of exclusive test is to improve the diagnostic capability of cause-effect diagnosis, it is also called *diagnostic test*. In this paper, we use "exclusive test" and "diagnostic test" interchangeably.

Recent papers [52, 53] describe an exclusive test generation system using existing automatic test pattern generation (ATPG) tools for single stuck-at faults. The authors define *diagnostic coverage* (DC) as a quantitative measure for the diagnostic capability of a given test set. They also provide algorithms for diagnostic simulation and exclusive test generation to measure and improve DC. The present work provides similar capability for diagnosis of transition delay faults (TDFs). Some ideas are similar [52, 53] but the advances are non-trivial.

We use modification of the circuit netlist (without actually modifying the actual circuit) to create a fault modeling netlist, which facilitates the detection and diagnosis of TDFs. The basic tool requirement for exclusive test is a test generation program for single stuck-at fault detection. In addition, test generation and fault simulation programs for TDFs are used to produce test vectors for all faults in a given set with required diagnostic coverage. Scan test environment is assumed in which both launch-off-capture (LOC) and launch-off-shift (LOS) types of tests can be applied. The principal objective is to go beyond the detection of TDFs and try to pinpoint the exact fault that may have caused the observed failure. We borrow the diagnostic coverage (DC) metric from previous work [52–55] to quantify the diagnostic capability of the final test set. Experimental results show that DC is significantly improved after employing the proposed automatic exclusive test generation. The new modeling technique and test generation system for transition delay faults are demonstrated to be simpler, more effective in distinguishing TDF pairs, and more time-efficient than previous work [18, 28, 29].

This paper is organized as follows. Section 2 starts with a flowchart of the cause-effect diagnosis, surveying previous work regarding each step of diagnosis. This section is intended to give the reader a complete picture of cause-effect diagnosis algorithm. In Section 3, we show how a transition delay fault on a line can be modeled by adding a logic gate and a flip-flop. Based on this modeling methodology, ATPG models for transition delay fault detection and diagnosis are described in Sections 4 and 5, respectively. In Section 5, we also propose an automatic exclusive test generation system. In Section 6, we briefly specify how scan test is generated under either LOC or LOS mode using Mentor Graphics Fastscan ATPG tool [32]. This is done for illustration and does not make the technique tool specific. This way we hope to give a clear view of how detection and exclusive tests are generated for transition delay faults in our experiments. Experimental results and comparison to previous work are given in Section 7. Section 8 concludes the paper.

2 Survey of Previous Work on Cause-Effect Fault Diagnosis

To survey previous work on cause-effect diagnosis algorithm, we first introduce a flowchart. Then, for each step we cite relevant work giving necessary details.

2.1 Flow of Cause-Effect Diagnosis Algorithm

A flowchart of cause-effect diagnosis is shown in Fig. 1. Initially, a complete set of fault detection tests or any high fault coverage test set is simulated by employing a diagnostic fault simulator. Meanwhile, a diagnostic dictionary is constructed and a diagnostic metric, such as diagnostic coverage (DC) discussed later in this section, is calculated. If the calculated metric of the initial test set is satisfactory, then good diagnostic capability is already achieved and no additional tests are needed. Otherwise, an exclusive test generator is utilized to generate diagnostic tests targeting the

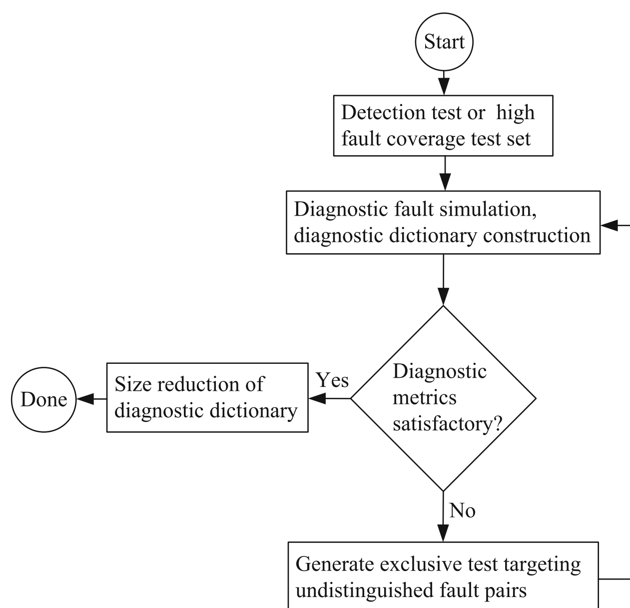


Fig. 1 Cause-effect diagnosis flow

undistinguished fault pairs left by the initial test set. Each new test is passed back to diagnostic fault simulator to identify more fault pairs which are accidentally distinguished by it. Each new test adds useful information to the diagnostic dictionary and the diagnostic metric is improved. The exclusive test generation and diagnostic fault simulation continues until the diagnostic metric reaches some expected values. The final diagnostic dictionary can be compressed using various techniques without losing diagnostic capability. During real silicon debug, candidate faults are selected based on comparison of the observed failures on tester against the stored expected responses in the dictionary.

2.2 Diagnostic Fault Simulation, Diagnostic Dictionary, and Diagnostic Metrics

Diagnostic fault simulation is a process that determines the diagnostic capability of a given set of tests with respect to a given set of faults and isolates indistinguishable fault pairs. It is widely known that diagnostic fault simulation based on equivalence classes [2, 11, 46, 47, 50–55] is more efficient than simulation considering all possible fault pairs explicitly [18, 19, 36, 37]. The equivalence class based diagnostic fault simulation works as follows.

Given a set of tests T, we group faults such that all faults within the same fault group, also called *equivalence class*, are not distinguishable from each other by those tests, while each fault in a group is pair-wise distinguishable from any fault from another group. A fault is said to be distinguished from another fault if there exists at least one test in T that produces different output on at least one observation point (a primary output or scan flip-flop), as determined by fault simulation of these two faults. The grouping of faults is similar to equivalence collapsing except that the grouping here is conditional to the test set. During equivalence class based diagnostic fault simulation, if a newly simulated test detects a subset of faults within a group then that group is partitioned into two groups, one containing the detected fault subset and the other containing the undetected fault subset. Based on the different output responses, the detected fault subset is further divided into different groups where faults with matching output responses are grouped together.

Prior to diagnostic fault simulation, all faults are put in a single group g_0. As the fault simulation of tests in T progresses, faults leave g_0 and start forming new groups. Suppose the set of tests T is sufficient to distinguish between every fault pair, then there will be as many groups as faults, and every group will contain just one fault.

A diagnostic dictionary for silicon debug is constructed during diagnostic fault simulation. Depending on the type of information compiled we may have a full-response dictionary or a pass/fail dictionary [45]. A full response dictionary contains all primary output (PO) specific information for each fault with respect to each test (also called *fault signature* [12] or *fault syndrome* [2]). Pass/fail dictionary is a compact form of dictionary that contains only pass or fail information for each test with respect to each fault.

Fault diagnosis is performed by matching the observed failures on tester to the expected response information in the dictionary. The complexity and cost of physical defect localization in real silicon largely depends on the size of the candidate fault set, size of the test set and diagnostic capability of the dictionary. Diagnostic fault simulation of different test sets will produce dictionaries that may have different diagnostic capability. Low diagnostic capability of a test set or a dictionary may induce diagnosis inefficiency or ambiguity.

Diagnostic capability has been quantified by several metrics. Some examples are diagnostic power (DP) [10], diagnostic resolution (DR) [2], diagnostic coverage (DC) [52–55], diagnostic expectation (DE) [40] and fault pair coverage (FPC) [10, 11, 47]. Their definitions follow.

$$DP = \frac{\text{number of fully distinguished faults}}{\text{total number of faults}} \quad (1)$$

where a fully distinguished fault is a fault that is distinguished from all other faults by the test set. A fully distinguished fault resides in a fault group only consisting of itself. A fault identified as fully distinguished is always dropped from further fault simulation, which is a common way to reduce diagnostic fault simulation time [11, 34, 47, 50, 51].

$$DR = \frac{\text{total number of faults}}{\text{total number of fault groups}} \quad (2)$$

$$DC = \frac{\text{total number of fault groups}}{\text{total number of faults}} \quad (3)$$

$$DE = \text{average fault group size} \quad (4)$$

$$FPC = \frac{\text{number of fault pairs distinguished}}{\text{total number of fault pairs}} \quad (5)$$

Note that DR is actually the reciprocal of DC. Note that FPC has also been referred to as diagnostic resolution [10, 11, 47]. Here we redefine it as fault pair coverage (FPC) to differentiate from the DR defined by Eq. 2 [2], because the two are not the same.

All diagnostic metrics provide information on diagnostic capability from different aspects. There is no simple way to classify them as better or worse. In this paper, we use DC and FPC to characterize the effectiveness of our exclusive test generation system. Obviously, given same set of faults and same set of tests, pass/fail dictionary has a lower diagnostic capability than its more comprehensive counterpart, the full-response dictionary. This is because the failing output information is not available in pass/fail dictionary.

For example, two faults detected by the same test but at different outputs are not distinguished by the pass/fail dictionary.

2.3 Methods of Speeding up Diagnostic Fault Simulation

The diagnostic fault simulation and dictionary construction can be computationally expensive. One way to speed up diagnostic fault simulation is to use *list* data structure to represent the equivalence classes of faults [47]. This reduces the required memory storage and facilitates fault dropping during simulation. A sequential diagnostic fault simulator and a distributed diagnostic fault simulator have been proposed [11] to speed up the equivalence-class based diagnostic fault simulation for synchronous sequential circuits. Another method [50] involves a simultaneous diagnostic fault simulation and equivalent fault pair identification so that equivalent faults are dropped early, thus speeding up fault simulation.

2.4 Methods of Exclusive Test Generation

Additional exclusive tests are necessary if the estimated value of diagnostic metric is not satisfactory. We broadly classify the existing exclusive test generation methods into three types. The first type, named *N-detection based* test generation, directly utilizes an N-detect test set for diagnostic purpose. In such a set, every fault is detected at least by N tests. N-detect tests have shown promising diagnostic ability [31, 48]. By increasing N, we expect that more faults including some timing related faults can be diagnosed. The problem of N-detection based methods is that the number of tests quickly grows with increasing N. Though methods have been proposed [24–26, 43–45] for compaction of N-detect test set, they are based on integer linear programming (ILP), which is NP-complete.

We call the second type of methods *algorithmic*. Genetic algorithms have been proposed [51] for generating exclusive tests for synchronous sequential circuits. However, a genetic algorithm based test generator is incapable of identifying equivalent fault pairs. A branch-and-bound algorithm, on the other hand, can distinguish specific pairs of faults in combinational circuits [15]. That method requires a specialized ATPG tool to be developed, which might incur high cost. Boolean satisfiability (SAT) based diagnostic test generation has been proposed for distinguishing transition delay fault pairs [6]. A SAT formulation becomes too complex for large circuits. Alternatively, a test elimination algorithm [35] generates exclusive tests for sequential circuits by randomly eliminating test vectors within a test sequence. By doing so, some faults are no longer detected by the test sequence and are thus distinguished from the other detected faults.

The third type of methods, also the focus of the present work, are called *ATPG-model based* exclusive test generation. An ATPG model is constructed for the circuit under test (CUT) whose netlist is modified to represent a target fault as a stuck-at fault. The modification amounts to insertion of a few logic elements for modeling purpose only. These methods temporarily modify the original circuit description so that a detection test of an inserted stuck-at fault becomes an exclusive test for a target fault pair. This type of methods are favored because they can readily utilize existing ATPG tools and guarantee to identify equivalent fault pairs assuming the run time limit of the ATPG tool is high enough.

An ATPG model for exclusive test generation for stuck-at faults transforms the problem to that of multiple-fault detection [2]. This ATPG model contains two copies of the circuit under test and requires an XOR gate to be added to each primary output. For a large circuit, the modeling complexity can be overwhelming. Later papers [46, 53] introduce another ATPG model for exclusive test generation. That model is simpler as it only requires one copy of the circuit, one additional primary input, and two additional logic gates.

The above methods target stuck-at faults. A basic assumption in the transition delay fault model is that the amount of fault induced delay in the signal transition is large, since the observation path may be, and often is, a short path [9]. Transition delay faults are not perfect and in fact may not represent many of the actual distributed timing defects [9, 21, 38]. Their acceptability, similar to that of stuck-at faults, is due to practical reasons. To name a few, their number grows only linearly with circuit size, they require two-vector tests that are essential for detecting timing related and other non-classical faults, and the scan methodology can be adapted to test them [9].

Diagnosis based on transition delay fault model helps diagnose real defects in silicon. For example, recent work [16] has shown how to transform stuck-open defect detection problem to transition delay fault detection problem. This equivalence means that diagnostic tests for transition delay faults can be also used to diagnose stuck-open defects, i.e., gate oxide defects of transistors. The present work aims at generating exclusive tests for transition delay faults. The correlation of transition delay fault model to real silicon defects is of practical interest. In this work, we hope to provide a similar diagnostic capability for transition delay faults as is generally available for stuck-at faults [2, 9, 15, 36, 37, 46, 50–53]. We propose an ATPG model to facilitate diagnostic test generation. Compared to a previous attempt [18] where the ATPG model consisted of up to four copies of the circuit under test, our ATPG model works without duplicating the circuit at all, although the model is mildly sequential. The sequential behavior comes from two modeling flip-flops. These flip-flops are pre-initialized and

hence do not increase the ATPG complexity. The proposed ATPG model is equally suitable for test generation under both LOC and LOS modes, which is another advantage over the previous work [18] that only supported the LOC mode. Our model also supports test generation using either sequential or combinational ATPG tool based on user's choice. For combinational ATPG tool, the CUT needs to be unrolled into two time-frames. Even so, it is still more efficient because only two, and not four [18], copies of CUT are required.

Another attempt [28, 29] at generating diagnostic tests for TDFs has similarity to this work. However, two 4-to-1 multiplexers and two copies of the circuit were required in that ATPG model, which made it more complex. In our model, only two 2-input logic gates and two flip-flops are required in a single copy of the circuit. The two inserted flip-flops make the circuit slightly sequential, requiring the use of a sequential ATPG tool for exclusive test generation without duplicating the circuit.

2.5 Size Reduction of Diagnostic Dictionary

A fault dictionary with good diagnostic capability usually contains huge amount of data increasing memory requirement, diagnosis time and cost. To reduce the dictionary size, two classes of methods are worth mentioning. The first class of methods [12, 27, 34] organize and encode the dictionary to reduce the size either without, or with little, reduction in diagnostic metric. The organization of a dictionary represents the ordering of its content. Dictionary encoding is a technique, which expresses fault signatures in a compact format to reduce memory storage requirement.

Another class of methods reduce dictionary size by reducing the total number of exclusive tests in the dictionary. Given fault signatures of all faults with respect to all tests, the diagnostic test set minimization problem can be transformed to a set covering problem and solved by constructing an integer linear programming (ILP) model. However, the huge amount of data contained in a dictionary requires a large number of constraints in the ILP model, which makes it computationally expensive. To reduce the complexity, Shukoor et al. [43–45] defined "generalized fault independence" to identify fault pairs that are guaranteed to be distinguished, and thus reduced the total number of constraints. Instead of solving a single ILP model, they also proposed a heuristic method of solving two less complex ILP models sequentially for exclusive test set minimization. Compared to a single complex ILP model, their method may result in slightly larger set of final exclusive tests, but with significantly reduced computational effort. Higami et al. [19] proposed heuristic methods for exclusive test set minimization as an alternative to the overwhelming set cover problem. The main idea is that they do not consider the entire dictionary all at once but consider a small dictionary constructed from a subset of fault pairs. A nearly minimal set of tests that distinguishes the subset of fault pairs is selected by solving a less complex set cover problem. The selected tests are then supplied to diagnostic fault simulation. After that, another subset of yet undistinguished fault pairs is chosen and a small dictionary is constructed again. The whole process terminates when all fault pairs distinguished by original test set are also distinguished by the reduced test set.

Note that after applying the second class of methods, i.e., diagnostic test vector minimization, the first class of methods, i.e., dictionary encoding and organization can be utilized to further reduce the dictionary size.

3 Modeling a Transition Delay Fault

A slow-to-rise or slow-to-fall transition delay fault on a line can be modeled by using a modeling flip-flop (MFF) and a logic gate (either an AND or an OR gate). This is illustrated in Fig. 2 where the shaded elements are inserted for modeling purpose and do not represent physical modification of the actual circuit. MFF is always initialized to a state that makes the circuit function correctly during the first vector. The state may then be changed by the first vector to a blocking signal that helps model the transition delay fault. For example, consider the slow-to-rise fault modeled in Fig. 2b. MFF is initialized to 1, ensuring that the output x' of line xx' will be the correct logic value in the first vector. Of the four vector-pairs on x, 00, 01, 10 and 11, all except 01 will produce the correct output at x'. The sequence 01 at x will

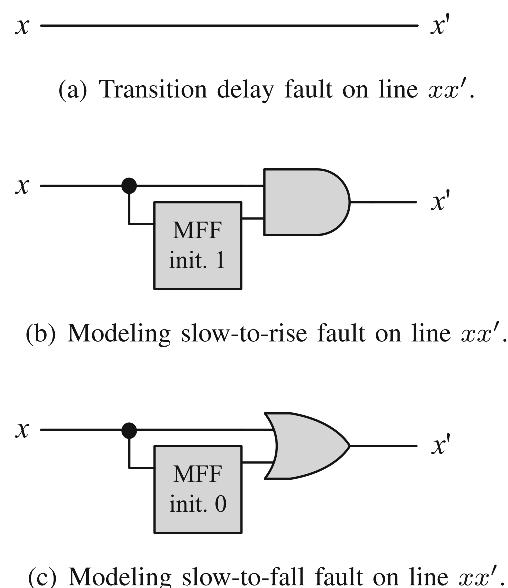


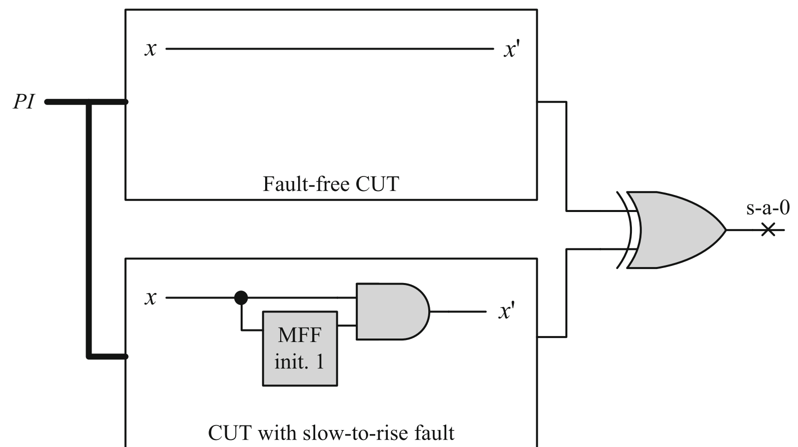
Fig. 2 Modeling a transition delay fault on line xx'

appear as 00 at x', properly representing a slow-to-rise transition fault on line xx'. Figure 2c shows a similar model for a slow-to-fall transition delay fault on line xx'.

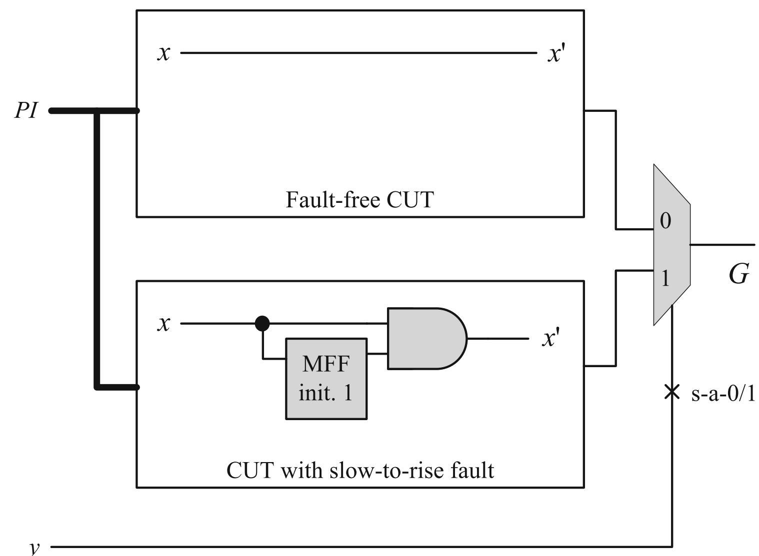
4 ATPG Model for Transition Delay Fault Detection

For the detection of a transition delay fault, we construct a single stuck-at fault ATPG model as shown in Fig. 3. The circuit of Fig. 3a is a miter circuit modeling the Boolean satisfiability formulation. Note that a logic "1" on output of the XOR gate cannot be obtained by a single vector. This is because the modeling flip-flop MFF is initialized to 1, setting $x' = x$, initially. Considering a two-vector test pattern, to produce a different output from the fault-free circuit the first vector must set $x = 0$ and then a second vector should set $x = 1$, besides sensitizing a path from x' to an observation point like a primary output (PO) or a scan flip-flop. So, a two-vector test pattern generated to detect the stuck-at-0 (s-a-0) at the output is actually a test for line xx' slow-to-rise transition delay fault. The ATPG model of Fig. 3b can be used in the same way. Any two-vector test for either s-a-0 or s-a-1 fault on y must produce different outputs for the fault-free and faulty circuits. An advantage of the model in Fig. 3b is that it can be simplified to use a single copy of

Fig. 3 ATPG model where a test for a single stuck-at fault is a test for xx' slow-to-rise transition delay fault



(a) An ATPG model: a test for output s-a-0 detects the slow-to-rise fault on xx'.



(b) An alternative ATPG model: a test for any stuck-at fault on y detects the slow-to-rise fault on xx'.

the circuit, which is an improvement over previous work [18, 28, 29].

The basis for collapsing the two copies of the circuit in Fig. 3b into a single copy as in Fig. 4 is the fact that the two copies of the circuit are almost identical, the only difference being at the faulty line. The multiplexer at PO can be actually moved to the fault site in a single copy of the circuit [52, 55]. This ATPG model is shown in Fig. 4. Here, a transition delay fault is to be detected on an interconnect from x (source) to x' (destination). The source signal x is made to fan out as two signals $x1$ and $x2$. Fan-out $x1$ is left as fault-free signal. The other fan-out $x2$ is modified based on Fig. 2 to model the existence of a transition delay fault. These two signals $x1$ and $x2$ are supplied to a 2-to-1 multiplexer whose output x' now feeds to all destinations of original signal x'. The control signal of this 2-to-1 multiplexer is a new primary input (PI) signal y. Any two-vector test detecting a stuck-at fault (either s-a-0 or s-a-1) on y will also detect the targeted TDF. Figure 4 shows the ATPG model for a slow-to-rise fault on xx', a slow-to-fall transition delay fault can be modeled similarly.

The gate and multiplexer combination in Fig. 4 is further simplified to the slow-to-rise ATPG model shown in Fig. 5a. The ATPG model for slow-to-fall transition delay fault is shown in Fig. 5b.

For simplicity of illustration, the derivation of ATPG models shown in Figs. 4 and 5 are for single-output circuit. However, the modeling technique in these two figures exactly applies to multiple-outputs circuit as well. Also, the ATPG models shown in Figs. 3, 4 and 5 are for a combinational circuit. However, the above fault modeling procedure exactly applies to scanned sequential circuits too. For either a scanned sequential circuit or combinational circuit under test, the existence of the modeling flip-flop (MFF) facilitates a two-vector test to be generated since the initial state of the flip-flop makes it impossible to activate the fault in a single vector. This two-vector test can be generated either by a sequential ATPG system in the partial scan mode to accommodate MFF or by a combinational ATPG tool

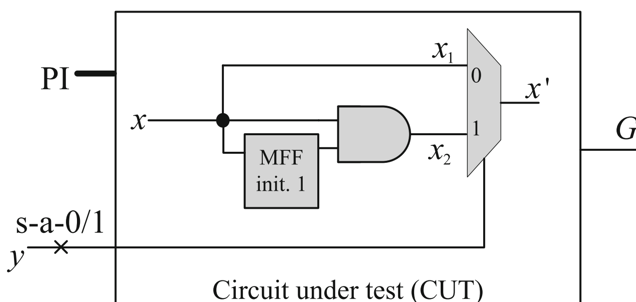
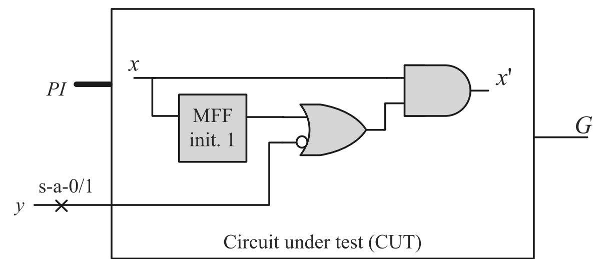
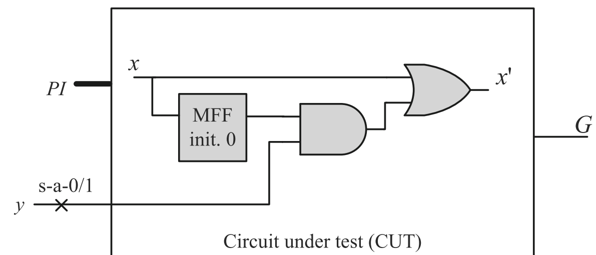


Fig. 4 A single-circuit-copy ATPG model where a test for stuck-at fault on y detects the slow-to-rise fault on line xx'



(a) ATPG model for slow-to-rise fault detection on line xx'.



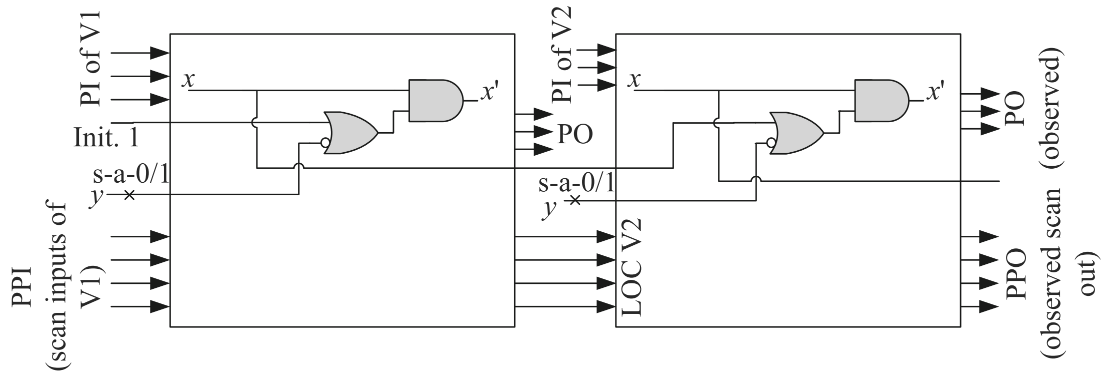
(b) ATPG model for slow-to-fall fault detection on line xx'.

Fig. 5 Simplified single-circuit-copy ATPG model where a test for a stuck-at fault on y detects the transition delay fault on line xx'

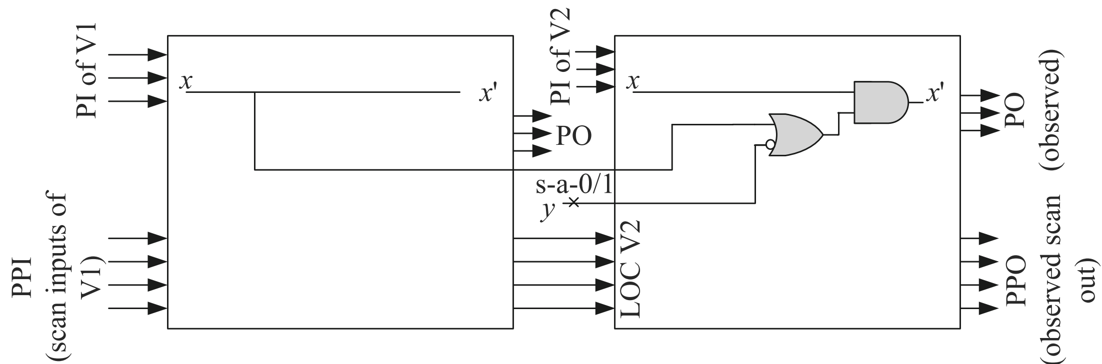
applied to a two-time-frame expansion of the ATPG model. For a scanned sequential circuit, the second vector can be generated for either LOC or LOS mode.

Figure 6a shows the two-time-frame expansion of the proposed ATPG model for a scanned sequential circuit where a two-vector test is generated under LOC mode. A slow-to-rise delay fault on line xx' is modeled using the construction of Fig. 5a. In the first time-frame, the initial state 1 of the unscanned MFF is applied through a fixed input shown as "Init. 1". All scan flip-flops (SFF) are stripped off and represented by pseudo primary inputs (PPI) and pseudo primary outputs (PPO) in the figure. Vector $V1$ consists of the normal PI and PPI. Vector $V2$ consists of PI in the second time-frame and PPI, which are actually PPO from the first time-frame. All outputs of the second time-frame are observable either directly as PO or through scan-out as PPO. Two stuck-at faults appear on the duplicated y input in the two time-frame circuit. A closer examination shows that it is impossible for the first stuck-at fault to cause any effect in the first time-frame due to the fixed "init. 1" input. Thus, the circuit can be further simplified as Fig. 6b with a single stuck-at fault on y for which any conventional combinational ATPG tool can be used to obtain a test. Since the state of MFF in the second time-frame is not observable and of no interest to us, we omit the output of MFF in Fig. 6b.

Similarly, Fig. 7a and b show two time-frame expansion of the proposed ATPG model for a scanned sequential



(a) Two-time-frame expansion ATPG model for LOC test generation of a slow-to-rise fault on line xx'.



(b) Simplified two-time-frame expansion ATPG model where a test for y s-a-0/1 is a LOC test for slow-to-rise fault on line xx'.

Fig. 6 Two-time-frame expansion ATPG model which facilitates the LOC test generation of slow-to-rise fault by using a combinational ATPG tool

circuit under LOS test generation mode. Again, a slow-to-rise delay fault on line xx' is modeled using the construction of Fig. 5a. The basic difference from the LOC model is in the way the PPI bits are obtained in the second time-frame. For LOS test these bits are obtained by a one-bit shift of the PPI bits of $V1$. Based on methodologies shown in Figs. 6 and 7, we can similarly model the detection of a slow-to-fall transition delay fault in a two time-frame expansion circuit under either LOC or LOS mode.

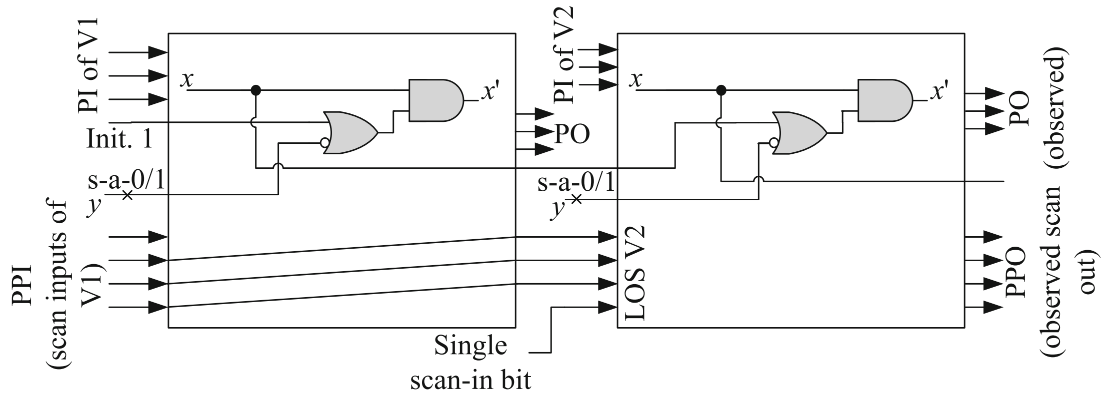
Somewhat similar ATPG model has been proposed before [28, 29] where authors use a 4-to-1 multiplexer to model a transition delay fault as a stuck-at fault. A close examination shows that their circuit can be simplified into the circuits of Figs. 6 and 7. In [28, 29], the model is used to generate single-observation (SO) single-location-at-a-time (SLAT) test [20] for each suspected fault so as to refine the final results of multiple fault diagnosis. Our ATPG model can be utilized for exactly the same purpose, however, with reduced modeling complexity.

5 Exclusive Test Generation for Transition Delay Faults

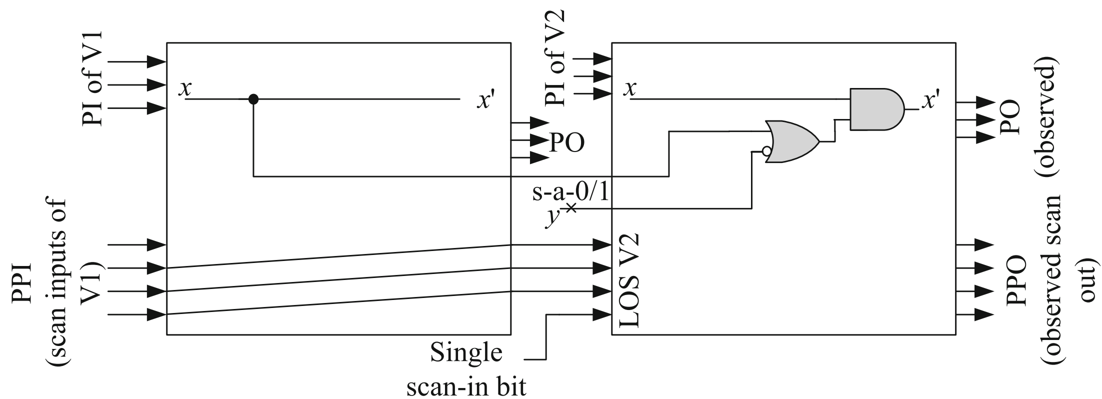
The previous section transforms the transition fault detection problem to a single stuck-at fault detection problem. That development will help us create an ATPG model for exclusive test generation, which is the focus of this work. In this section, we expand the scope of this work to build an automatic diagnostic test generation system for transition delay faults.

5.1 ATPG Model for Exclusive Test Generation

Suppose there are two transition delay faults $f1$ and $f2$. Without loss of generality, we assume that $f1$ is a slow-to-fall fault on line $x1x1'$ and $f2$ is a slow-to-rise fault on line $x2x2'$. Our ATPG model for exclusive test generation is shown in Fig. 8 where a two-vector test for stuck-at fault on y is an exclusive test for $f1$ and $f2$. For any other combination of transition delay fault pair, a similar structure can be constructed.



(a) Two-time-frame expansion ATPG model for LOS test generation of a slow-to-rise fault on line xx'.



(b) Simplified two-time-frame expansion ATPG model where a test for y s-a-0/1 is a LOS test for slow-to-rise fault on line xx'.

Fig. 7 Two-time-frame expansion ATPG model which facilitates the LOS test generation of slow-to-rise fault by using a combinational ATPG tool

By setting the maximum sequential depth in ATPG tool as 2, a detection test for either y s-a-0 or s-a-1 in Fig. 8, if it exists, will always contain two vectors. The second vector of this exclusive test can be generated either as an LOC sequence or as an LOS sequence for a full-scan sequential circuit. Figure 9 shows the two time-frame expansion of the circuit in Fig. 8. For simplicity, here we only consider the test generation for LOC mode. Similar considerations can be applied to test generation for LOS mode (refer to Fig. 7).

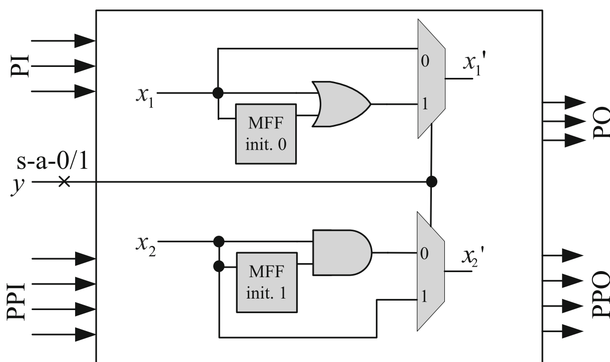


Fig. 8 ATPG model for exclusive test generation

The circuit of Fig. 9 has two faults. However, it is impossible for the first stuck-at fault to cause any observable effect. The simplified version is shown in Fig. 10. We use Fig. 10 to establish the correctness of the proposed ATPG model for exclusive test generation.

5.2 Proof of Correctness of the ATPG Model for Exclusive Test Generation

If a test vector pair distinguishes between a transition delay fault pair (say, faults $f1$ and $f2$), one of the following conditions should be satisfied [18]:

$C1$) $f1$ is detected and $f2$ is not detected.
$C2$) $f2$ is detected and $f1$ is not detected.

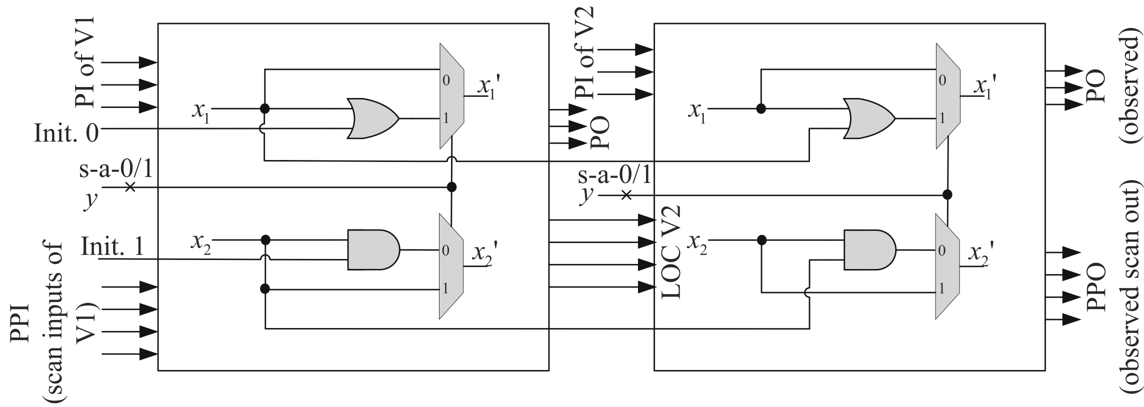


Fig. 9 Two-time-frame expansion ATPG model for exclusive test generation under LOC mode

$C3$) $f1$ and $f2$ are both detected, but there exists at least one observation point where only one fault effect is propagated.

For a single transition delay fault on line l that is slow to change from w to $\overline{w}$ not being detected by a vector pair $(V1,\ V2)$, at least one of the following conditions must be true:

$U1$) $\overline{w}$ is set on l by $V1$ (fault not activated).
$U2$) w is set on l by $V2$ (fault not activated).
$U3$) stuck-at w fault on l is not propagated by $V2$ to an observation point (fault activated but not propagated).

Hence, to distinguish between two transition delay faults, either one of the following conditions should be satisfied:

$D1$) Generate a vector pair satisfying $C1$ or $C2$, and one fault is not detected due to $U1$.
$D2$) Generate a vector pair satisfying $C1$ or $C2$, and one fault is not detected due to $U2$..
$D3$) Generate a vector pair satisfying $C1$ or $C2$, and one fault is not detected due to $U3$.
$D4$) Generate a vector pair satisfying $C3$.

In Fig. 10, for a two-vector test generated by an ATPG tool to detect the stuck-at fault on y, the first vector $V1$ must satisfy either one of the following conditions:

$A1$) $V1$ sets $x1$ =1, $x2$ = 0, which means $V1$ activates both $f1$ and $f2$. Then, the second time-frame of Fig. 10 is reproduced in Fig. 11. According to [53], the detection test ($V2$ here) for stuck at fault on y is an exclusive test for $x1$ s-a-1 and $x2$ s-a-0. $V2$ can distinguish $x1$ s-a-1 and $x2$ s-a-0 in either of the following three cases:

1) $V2$ detects $x1$ s-a-1 but but does not detect $x2$ s-a-0 ($C1$).
2) $V2$ detects $x2$ s-a-0 but does not detect $x1$ s-a-1 ($C2$).
3) $V2$ detects both $x1$ s-a-1 and $x2$ s-a-0, but there is at least one observation point where only one fault effect is propagated ($C3$).

The non-detection of a fault here can be due to either $U2$ or $U3$. We conclude that condition $A1$ incorporates conditions $D2$, $D3$, and $D4$.

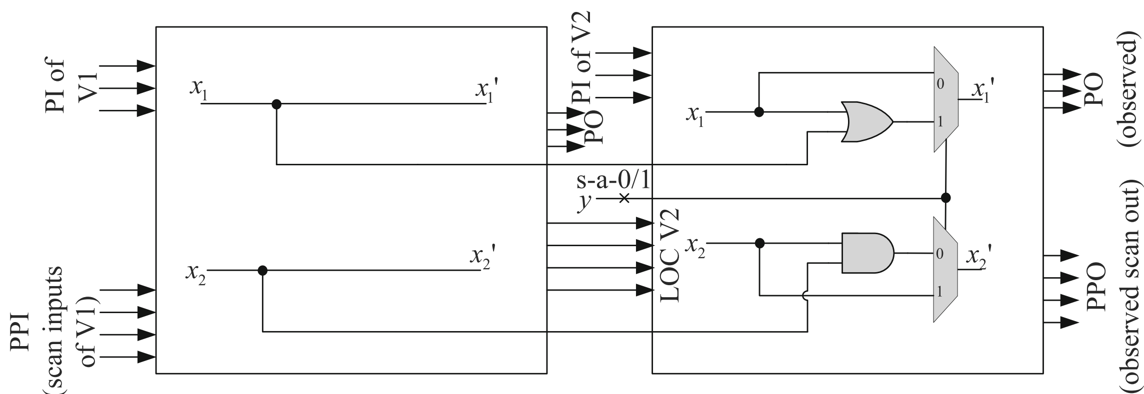


Fig. 10 Simplified two-time-frame expansion ATPG model for exclusive test generation under LOC mode

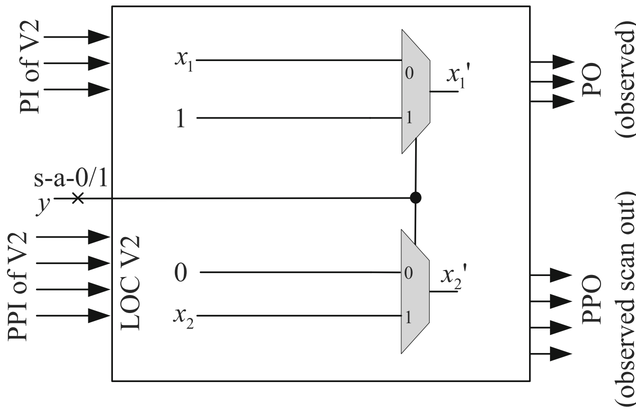


Fig. 11 Second time-frame under condition $A1$

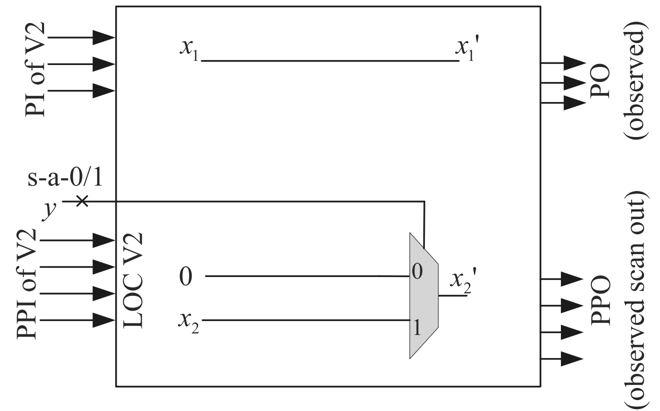


Fig. 13 Second time-frame under condition $A3$

$A2$) $V1$ sets $x1 = 1$, $x2 = 1$, which means $V1$ activates $f1$ only. Then, the second time-frame looks like Fig. 12. Based on definition of Boolean difference [9], the second vector $V2$ generated by ATPG for detection of stuck-at fault on y is a detection test for $x1$ s-a-1. In this scenario, $f1$ is detected by the two-vector test and $f2$ is not detected (condition $C1$). The non-detection of $f2$ is due to condition $U1$. Note that condition $A2$ incorporates $C1$ with $U1$.

$A3$) $V1$ sets $x1 = 0, x2 = 0$, which means $V1$ can activate $f2$ but not $f1$. Then, the second time-frame looks like Fig. 13. Obviously, the second vector $V2$ is a detection test for $x2$ s-a-0. In this scenario, fault $f2$ is detected but $f1$ is not detected (condition $C2$). The non-detection of $f1$ is due to condition $U2$. Note that condition $A3$ incorporates $C2$ with $U1$. $A2$ and $A3$ together incorporate condition $D1$.

If $x1 = 0$ and $x2 = 1$ are set by $V1$, then both TDFs will not be activated, and an exclusive test will be impossible. An ATPG tool will avoid this condition, automatically.

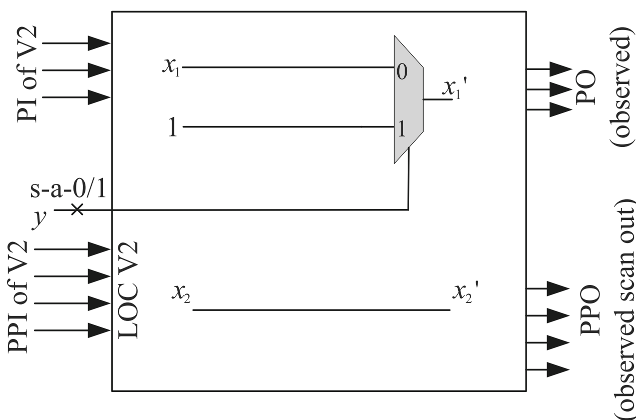


Fig. 12 Second time-frame under condition $A2$

With the proposed ATPG model, a two-vector test for a single stuck-at fault will always satisfy any condition from $D1$ to $D4$, and thus distinguish the targeted TDF pair. A similar proof can be derived for test generation under LOS mode, and for other combinations of transition delay faults. If the modeled stuck-at fault is proven redundant by the ATPG tool, then the targeted TDFs are equivalent.

5.3 A Diagnostic Test Generation System

We devise an automatic exclusive test generation system to generate both detection and diagnostic tests for a given circuit to improve the diagnostic coverage (DC). This system is used for test generation in our experiments and its flowchart is shown in Fig. 14. The whole system is

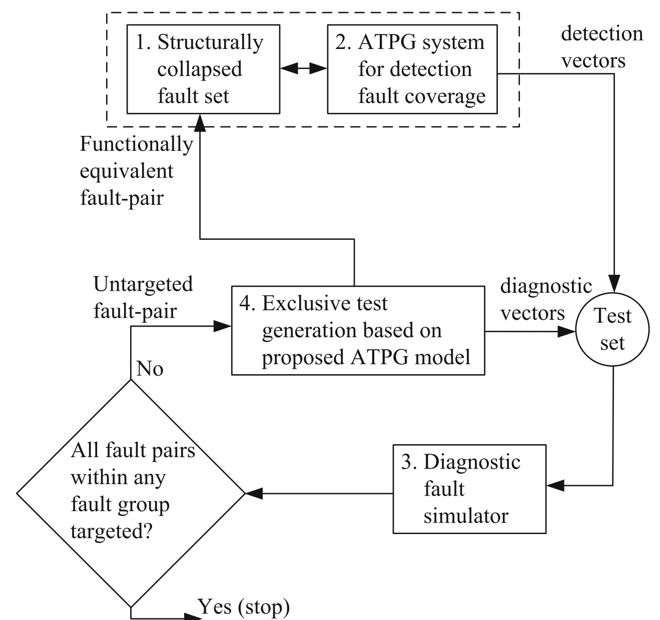


Fig. 14 Flowchart of automatic exclusive test generation system

implemented in Python programming language [39] and consists of four functional blocks. Blocks 1 and 2 form a conventional detection ATPG system. Block 4 does exclusive test generation based on the proposed ATPG model of Fig. 8. Block 3 is a diagnostic fault simulator.

In Fig. 14, after diagnostic fault simulation of detection tests, the faults are divided into groups. The ATPG model of Fig. 8 is utilized to generate exclusive tests targeting fault pairs within all fault groups. Diagnostic fault simulation is conducted after the generation of each exclusive test to update fault groups. If no exclusive test is generated by the ATPG tool for a fault pair then there can be two reasons. First, the modeled stuck-at fault is identified as redundant and no test is possible. In this case, the two faults are equivalent and we drop the first (or any one) fault from its corresponding fault group and from the fault list as well. Second, run was aborted due to search limit of ATPG tool. In this case, the target TDF pair remains indistinguishable and the diagnostic coverage remains unchanged. The aborted fault pair will be excluded from further exclusive test generation. The exclusive test generation process stops when all fault pairs within all fault group have been targeted for exclusive test generation.

6 Diagnostic Test Generation Using Fastscan

Although Mentor's Fastscan [32] is used for experimental illustration, any other system of test tools can be used similarly. For a sequential circuit under test, we first modify the circuit for scan testing using DFTAdvisor [33]. Thus, scan-in, scan-out and scan-enable (SE) signals are added to make it a full scan circuit. For an originally combinational circuit under test, we leave it as is. ATPG tools are then employed to generate detection and exclusive tests.

For detection test generation for transition delay faults, our ATPG models allow the tests to be generated by targeting a modeled single stuck-at fault. An ATPG tool like Mentor's Fastscan [32], however, can directly generate detection tests as well as simulate them for transition delay faults. In our experiments, we use that capability of Fastscan. For a scanned sequential circuit, the detection test generated for transition delay faults consists of a scan-in sequence and two primary input vectors. The application sequence of this test is: (1) Set scan enable SE = 1 for scan mode and apply the scan-in sequence, (2) Apply the first primary input vector, (3) Set SE = 0 (normal mode), apply second primary input vector and clock the circuit at the same time to launch the second vector, (4) Clock the circuit again with functional clock, which may be faster than the shift clock, to capture responses in scan flip-flops, and (5) Set SE = 1 and scan out the captured results. This sequence will detect the transition fault in the LOC mode.

For LOS mode, step (3) is modified as follows: we hold SE = 1 (scan mode) for one cycle, so the circuit will be clocked in the scan mode for one clock period while new primary inputs are applied. The other steps remain the same. One disadvantage of LOS mode is that it requires a global scan-enable signal. All scan flip-flops need to change mode (typically, scan to normal) synchronized with a high speed test clock. In general, it is difficult to implement a high speed global scan-enable signal though specialized test solutions have been evolved [3, 4]. However, in general, LOS is often not used in spite of its better fault coverage than that of LOC. In our experiment, detection tests for scanned sequential circuits are generated under LOC mode only. For a combinational circuit, the detection test will only contain two PI vectors and there is no need to specify LOC or LOS option to Fastscan [32].

For exclusive test generation, Fastscan allows test generation in the partial scan mode provided the number of non-scan flip-flops is small. That capability is useful for the ATPG model of Fig. 8, which contains two non-scan flip-flops. We use that capability of Fastscan [32]. The MFFs are treated as pre-initialized non-scan flip-flops. All scan flip-flops and MFFs are connected to the same clock source. To distinguish a transition fault pair, Fastscan then generates a two-vector test for a single stuck-at fault. For a scanned sequential circuit, this test is generated under LOC mode in our experiments. An exclusive test for a TDF-pair in a purely combinational circuit contains just two PI vectors.

7 Experimental Results

7.1 Exclusive Test Generation for Combinational and Sequential Benchmark Circuits

We run the automatic exclusive test generation system on scanned ISCAS'89 [7] sequential benchmark circuits and ISCAS'85 [8] combinational benchmark circuits. The results are shown in Tables 1 and 2, respectively.

The first column of Table 1 shows the circuit name. The second column lists the number of transition delay faults (TDFs). An input fault on a fan-out free interconnect or the input of a NOT gate is collapsed with the output fault of the driving gate since no exclusive test exists for such structurally equivalent faults. Also, some redundant faults are identified by Fastscan [32] in the detection phase and they are removed as well. This is because all faults in the set of redundant faults are neither detectable not distinguishable from each other. We also excluded all aborted faults since aborted faults are likely to be aborted again during exclusive test generation [2]. Thus, we start with structurally collapsed

Table 1 Detection and diagnostic test generation for transition delay faults (TDFs) in full-scan ISCAS'89 sequential benchmark circuits

Circuit	Number of TDFs	Detection test generation: Number of det. tests	*FC* (%)	*DC* (%)	No. of Undiag. groups	Largest group size	No. of Undiag. pairs	Diagnostic test generation: No. of Diag. tests [52, 55]	No. of Diag. tests	*DC* (%)	No. of Undiag. groups	Largest group size	No. of Undiag. pairs	CPU time (sec.)*
s27	46	11	100.0	52.2	12	7	45	18	17	97.8	1	2	1	29
s298	385	44	79.9	62.4	62	5	111	34	26	70.1	39	4	47	40
s382	498	51	80.8	64.1	82	4	126	24	17	67.9	61	4	80	55
s1423	2198	102	92.0	79.3	280	5	380	106	90	84.2	182	5	208	845
s5378	6007	208	90.5	82.2	374	10	1010	472	395	89.6	85	7	99	488
s9234	10567	355	93.8	86.2	968	11	2033	597	592	94.1	391	4	549	1864
s13207	13006	497	87.8	69.8	1742	15	4942	543	490	74.1	1411	8	2184	3499
s15850	15340	312	84.7	71.3	1945	10	4300	486	389	74.3	1594	9	2325	3927
s35932	52414	73	87.3	88.4	3622	7	8049	725	662	90.2	2899	4	4611	13835
s38417	47114	237	98.0	87.4	4131	11	6820	1336	1179	90.8	2900	8	3613	15341
s38584	53823	396	92.5	86.6	4051	9	6676	1793	1645	90.3	2506	6	3056	14841

*Hardware configuration: 2 × 1.0GHz CPU, 1885 MB RAM, x86 Linux

Table 2 Detection and diagnostic test generation for transition delay faults (TDFs) in ISCAS'85 combinational benchmark circuits

Circuit	Number of TDFs	Detection test generation: Number of det. tests	*FC* (%)	*DC* (%)	No. of Undiag. groups	Largest group size	No. of Undiag. pairs	Diagnostic test generation: No. of Diag. tests	*DC* (%)	No. of Undiag. groups	Largest group size	No. of Undiag. pairs
c17	34	7	100.0	73.5	6	3	12	8	100.0	0	0	0
c432	774	73	98.7	79.3	85	8	307	92	99.6	3	2	3
c499	910	106	99.1	91.2	55	4	108	64	100.0	0	0	0
c1355	2553	174	99.5	84.3	229	6	651	132	91.7	104	3	208
c1908	2929	184	99.7	89.8	173	6	533	201	99.9	2	3	3
c2670	4151	106	96.4	89.2	271	10	678	139	96.7	63	10	96
c3540	5459	230	96.6	87.8	470	5	885	329	97.3	100	4	161
c5315	8777	135	99.3	96.0	264	11	499	176	99.7	11	5	18
c6288	12409	60	99.2	94.6	596	4	645	25	95.0	556	3	575
c7552	12135	168	98.8	94.1	513	10	1073	296	98.6	130	7	159

faults and consider only faults detectable by the detection test set for diagnosis.

The third column of Table 1 lists the number of LOC tests generated by Fastscan for fault detection. The fault coverage (FC) of detection test set is given in column 4. Reasons for less than 100 % FC are: (a) aborted ATPG runs and (b) LOS mode not used. An ATPG run is aborted due to some preset per fault limit on computation effort. The faults left undetected may also include some redundant faults. Column 5 of Table 1 gives the diagnostic coverage (DC) of the detection test set obtained from diagnostic fault simulation. As can be seen, circuit s27 has 12 undiagnosed fault groups and the largest group contains 7 faults (see columns 6 and 7). An *undiagnosed fault group* here refers to a fault group containing more than one fault. Similarly, circuit s5378 has 374 undiagnosed groups, the largest one containing 10 faults. Column 8 shows the total number of indistinguishable fault pairs after diagnostic fault simulation of detection test set. Additional exclusive tests targeting fault pairs formed within each multi-fault group are generated and their number is listed in column 10 of Table 1.

It is interesting to observe that for diagnostic ATPG if we target pairs of faults that are structurally near each other, then the total number of final exclusive tests will be smaller. This is because faults located structurally close to each other are typically hard to distinguish, while the faults located farther apart in the netlist are relatively easy to distinguish and are mostly distinguished by the tests that are derived for adjacent fault pairs. Similar observations were described by other authors [36] where subsets of neighborhood faults determined from structural analysis of the circuit netlist are used to reduce the number of target fault pairs during diagnostic test generation. We used this heuristic to obtain a smaller set of exclusive tests. Compared to previous results [52, 55] shown in column 9, we notice the reduction in the number of final exclusive tests. However, our focus in this work is on effective and efficient exclusive test generation and a comprehensive compaction of exclusive test patterns is beyond the scope of this work. An interested reader may refer to [19, 43–45] for more details on diagnostic test set compaction.

The diagnostic coverage (DC) of total test set is given in column 11. For example, 17 exclusive tests were generated for s27 raising DC from 52.2 % to 97.8 %. There was only one undiagnosed fault group left (column 12) and it contained two faults (column 13). Column 14 shows the total number of remaining indistinguishable fault pairs. Column 15 shows the CPU time for diagnostic test generation and corresponding diagnostic fault simulation. As can be seen, the CPU time increases as the circuit complexity increases. Since diagnostic test generation and dictionary construction are all one-time tasks, and moreover the diagnostic coverage is largely improved by exclusive test generation, the listed CPU time may be considered acceptable.

Columns in Table 2 have the same meaning as those in Table 1 except that exclusive tests of previous work [52, 55] were not available for the combinational benchmark circuits. Although CPU times for all circuits are not recorded, the circuit requiring most CPU time in diagnostic test generation and fault simulation was c7552, which consumed 2,327 seconds on the processor identified under Table 1. The CPU time for detection test generation and diagnostic fault simulation is always much less than that of diagnostic test generation and diagnostic fault simulation.

One observation made from Tables 1 and 2 is that the proposed automatic exclusive test generation system always improves DC beyond that of detection test set. It also helps reduce the number of undiagnosed fault groups, the largest undiagnosed group size, and the total number of indistinguishable fault pairs. These observations demonstrate the effectiveness of the proposed system. Compared to sequential benchmark circuits, the FC and DC are always higher for combinational benchmark circuits. This is because the test for sequential circuits are generated in LOC mode. In a two-vector LOC test, the generation of the scan part of the second vector is always restricted by the scan part of the first vector and the function of the circuit. For combinational circuits, the second vector is generated with full flexibility since it only contains PIs. Because of higher flexibility in both detection test and exclusive test generation, the FC and DC are always higher for combinational benchmark circuits.

Another point worth mentioning is that Fastscan [32] for exclusive test generation operates in sequential mode (refer to Fig. 8). Due to some default backtrack limit used in this work, it often failed to identify redundancies even if the target fault pair was functionally equivalent. By utilizing the combinational two time-frame ATPG model of Fig. 10, it may be much easier for Fastscan to identify redundancy even with its default backtrack limit. If all fault equivalences could be identified, then diagnostic coverage will be further improved.

Actually, based on observations made from several small ISCAS'89 circuits through detailed structural analysis we found that many aborted fault pairs are actually functionally equivalent. The two undiagnosed faults of s27 are shown in Fig. 15. Using a two-time-frame combinational ATPG model, we determined that these two faults cannot be distinguished by any LOC test. Because the functional operation of the circuit is constrained to a subset of conditions allowed during the LOC testing, i.e., these two faults can be considered functionally equivalent. We have verified that with an added LOS test all faults in s27 can be distinguished. That will make $DC = 100$ %. Research should

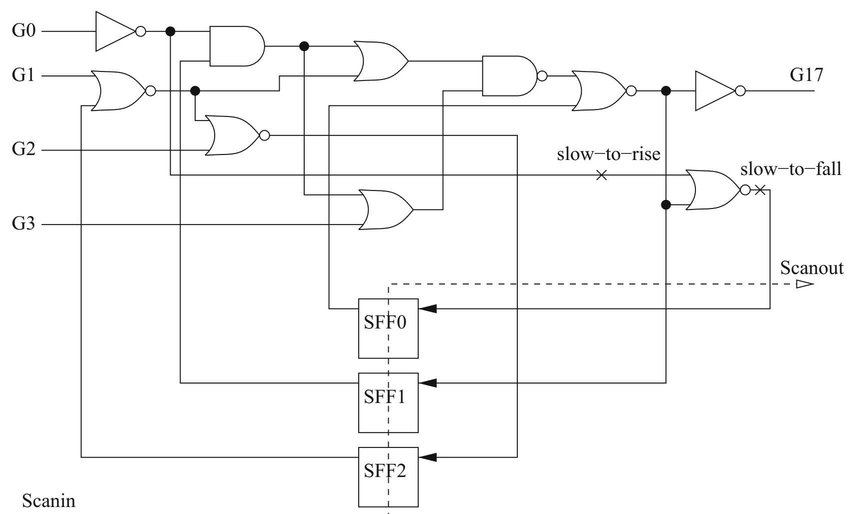


Fig. 15 Illustration of a fault pair indistinguishable in LOC mode of s27 benchmark circuit

be directed toward constructing a two time-frame combinational ATPG model under both LOC and LOS modes, so that a nearly 100 % DC could be expected for all sequential and combinational circuits.

7.2 Comparison to Previous Work

While authors of [28] and [29] utilized their ATPG model to generate Single Observation-Single Location At a Time (SO-SLAT) tests for multiple transition faults diagnosis, they actually assume a high-resolution diagnostic dictionary has already been constructed and initial diagnosis has been done. Their target is a small set of suspected TDFs returned by initial diagnosis. Because the purpose and application of their ATPG model is different from that of the proposed ATPG model, we did not compare our ATPG model to theirs in this section.

Another strategy [18] aims to generate additional diagnostic tests beyond detection tests so as to improve the diagnostic metrics of the diagnostic dictionary, which is also the goal of this work. Thus, a detailed comparison between our work and [18] is shown in Table 3. In [18], for the ISCAS'89 benchmark circuits, instead of targeting all transition delay faults, the authors randomly choose 1,000 faults and extract those pairs that cannot be distinguished by the detection test set. These pairs then serve as the starting set for their diagnostic test generation flow. In our case, targets for detection test set are all collapsed faults and the targets for diagnostic test generation are all remaining undistinguished fault pairs. Since only the number of distinguished fault pairs is reported in [18], we compare our work with theirs using the fault pair coverage (FPC) metric.

FPC in Table 3 is calculated with respect to diagnostic test set only. Columns 1 through 6 show, respectively, circuit name, number of target fault pairs for exclusive test generation, number of exclusive tests, number of distinguished fault pairs, FPC, and CPU time. For each circuit, the first row marked as (A) shows the results from [18] and the second row marked as (B) shows the results of this work. The last two rows of the table show the average value of the results obtained in [18] and in this work, respectively.

From Table 3, the FPC of this work is higher than that of [18] for most circuits. Only exception is circuit s38584 for which our FPC is 47.0 % while theirs is larger at 60.9 %. The average FPC achieved in this work is 58.4 % which is significantly higher than 15.6 % reported for [18].

Table 3 Comparing previous work [18] (A) and this work (B)

Circuit	No. of target pairs	No. of exclusive tests	No. of dist. pairs	FPC (%)	CPU time (sec.)*
s5378	(A) 7152	552	928	13.0	2916
	(B) 1010	395	911	90.2	488
s9234	(A) 8010	542	861	10.7	3094
	(B) 2033	592	1484	73.0	1864
s13207	(A) 5366	463	700	13.0	4997
	(B) 4942	490	2758	55.8	3499
s15850	(A) 15059	214	348	2.3	18014
	(B) 4300	389	1975	45.9	3927
s35932	(A) 9874	48	140	0.1	30045
	(B) 8049	662	3438	42.7	13835
s38417	(A) 281	163	171	60.9	891
	(B) 6820	1179	3207	47.0	15341
s38584	(A) 14197	583	1316	9.3	174649
	(B) 6676	1645	3620	54.2	14841
Average	(A) 8562.7	366.4	637.7	15.6	33515
	(B) 4832.9	764.6	2484.7	58.4	7685

*(A) Hardware configuration is not reported in [18]
(B) 2 × 1.0GHz CPU, 1885 MB RAM, ×86 Linux

Higher FPC indicates that much larger percentage of fault pairs are distinguished. One may argue that the fault pairs for which no LOC test were generated must be equivalent (indistinguishable) in [18] and their FPC should be 100 %. This is not true because of the following reasons:

1) For experiments in [18], it is assumed that values of primary inputs remain the same between the first and the second time-frame for the LOC exclusive test generation. This assumption will leave many fault pairs as indistinguishable although some can be actually distinguished. The big difference between the FPC in this work and [18] indicates that a large portion of fault pairs can be further distinguished if the primary input vector can be assigned values independently in the first and second time-frames.
2) Fault pairs that cannot be distinguished in LOC mode may still be distinguishable in the LOS mode.

The CPU configuration was not reported in [18] and therefore a true comparison is not possible. Consider the circuit s38584. The CPU time for our system is 14841 seconds for 1645 exclusive tests. Note that the CPU time in our work is not optimized. For every fault pair, Fastscan [32] flattens the CUT and reconstructs the data structure for test generation. Through experiments we find that nearly 99.8 % percent of the time is spent in data structure reconstruction. This huge overhead can be almost completely eliminated by modifying the data structure incrementally instead of constructing it from scratch every time. Once that is done, we believe the CPU time for the diagnostic test generation process will reduce to minutes for large circuits and seconds for small circuits. Thus the proposed system could have significant potential for high time efficiency.

8 Conclusion

Delay fault diagnosis is important characterizing the performance of modern VLSI devices and a high diagnostic coverage of tests is highly desirable. The ATPG models of transition delay faults presented in this work are significantly simpler than those previously published [18, 28, 29]. Also, combined with diagnostic fault simulation and automatic test generation, an automatic exclusive test generation system is demonstrated to be effective and efficient in improving the diagnostic coverage for various benchmark circuits.

The problem of detecting a path delay fault has also been modeled as detection of a single stuck-at fault [1]. Equipped with such a technique, we can easily construct an ATPG models to generate exclusive tests for path delay faults using the methodology proposed in this work. Generally speaking, for any fault whose behavior can be mapped onto single stuck-at faults with additional modeling logic gate(s), a conventional stuck-at fault ATPG tool can be used for detection or diagnostic test generation. Delay effects can be modeled with flip-flops and primitive logic gates (AND, OR, NOT, etc.) Multiplexers are added to choose between fault free behavior and faulty behavior (detection test generation) or between two faulty behaviors (exclusive test generation). Since the two faulty behaviors do not need to come from the same fault model, mixed fault model diagnosis is also possible using the methodology proposed in this work. For example, one could derive a test to distinguish between a stuck-at fault and a transition delay fault.

The exclusive tests generated by our test generation system are fully specified tests and may not be directly applied to circuit with test compression structure. However, there is existing work on identifying don't care bits in fully specified tests and utilize those bits to statistically encode the tests for compression [22, 23]. By combining these techniques with our test generation system, transition fault diagnosis can be done in test compression environment.

In the area of diagnostic test generation, whether or not the tests have an adequate diagnostic coverage cannot be ascertained unless we have an effective and efficient tool for identifying fault equivalence [5, 13, 14, 17, 41, 42, 50]. The present work provides the possibility of identifying equivalent faults by conventional ATPG tools. However, to identify each equivalent fault pair, we need to largely increase the backtrack limit and test generation time of ATPG tools. Future research must find more efficient ways for equivalence identification.

Another direction for future research is to make diagnostic tests specific to small delay defects (SDD), i.e., derive timing-aware test patterns to detect transition delay faults through longest sensitizable paths, i.e., paths with smallest slack from the fault site to an observable point [30].

Acknowledgments Research supported in part by the National Science Foundation Grants CNS-0708962 and CCF-1116213. Y. Zhang is with Broadcom Corporation, San Diego, CA 92127, USA.

References

1. Agrawal P, Agrawal V, Seth S. (1993) Generating tests for delay faults in nonscan circuits. IEEE Des. Test Comput. 10(1):20–28
2. Agrawal VD, Baik DH, Kim YC, Saluja KK (2003) Exclusive test and its applications to fault diagnosis. In: Proc. 16th international conf. VLSI design, pp 143–148
3. Ahmed N, Ravikumar CP, Tehranipoor M, Plusquellic J (2005) At-speed transition fault testing with low speed scan enable. In: Proc. 23rd IEEE VLSI test symposium, pp 42–47
4. Ahmed N, Tehranipoor M, Ravikumar CP, Butler KM (2007) Local at-speed scan enable generation for transition fault testing using low-cost testers. IEEE Trans Comput-Aided Des 26(5):896–906

5. Amyeen ME, Fuchs WK, Pomeranz I, Boppana V (2003) Fault equivalence identification in combinational circuits using implication and evaluation techniques. IEEE Trans Comput-Aided Des 22(7):922–936
6. Bhatti NK, Blanton RD (2006) Diagnostic test generation for arbitrary faults. In: Proc. international test conf., pp. 1–9. Paper 19.2
7. Brglez F, Bryan D, Kozminski K (1989) Combinational profiles of sequential benchmark circuits. In: Proc. IEEE international symposium on circuits and systems, pp 1929–1934
8. Brglez F, Fujiwara H (1985) A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In: Proc. IEEE international symposium on circuits and systems, pp 677–692
9. Bushnell ML, Agrawal VD (2000) Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. Springer
10. Camurati P, Medine D, Prinetto P, Reorda MS (1990) A diagnostic test pattern generation algorithm. In: Proc. international test conf., pp 52–58
11. Chen SC, Jou JM (1997) Diagnostic fault simulation for synchronous sequential circuits. IEEE Trans Comput-Aided Des 16(3):299–308
12. Chess B, Larrabee T (1999) Creating small fault dictionaries. IEEE Trans Comput-Aided Des 18(3):346–356
13. Doshi AS (2006) Independence fault collapsing and concurrent test generation. Master's thesis, Auburn University, ECE Dept.
14. Doshi AS, Agrawal VD (2005) Independence fault collapsing. In: Proc. 9th VLSI design and test symp., 357–364
15. Gruning T, Mahlstedt U, Koopmeiners H (1991) DIATEST: a fast diagnostic test pattern generator for combinational circuits. In: Proc. IEEE international conf. computer-aided design, pp 194–197
16. Han C, Singh A (2014) On the testing of hazard activated open defects. In: Proc. international test conf., pp 1–6, Paper 1.2
17. Hartanto I, Boppana V, Fuchs WK (1996) Diagnostic fault equivalence identification using redundancy information & structural analysis. In: Proc. international test conf., pp 20–25
18. Higami Y, Kurose Y, Ohno S, Yamaoka H, Takahashi H, Takamatsu Y, Shimizu Y, Aikyo T (2009) Diagnostic test generation for transition faults using a stuck-at ATPG tool. In: Proc. international test conf., pp 1–9, Paper 16.3
19. Higami Y, Saluja KK, Takahashi H, Kobayashi SY, Takamatsu Y (2006) Compaction of pass/fail-based diagnostic test vectors for combinational and sequential circuits. In: Proc. Asia and South Pacific conference on design automation, pp 659–664
20. Huisman LM (2006) Diagnosing arbitrary defects in logic designs using single location at a time (SLAT). IEEE Trans Comput-Aid Des 23(1):91–101
21. Iyengar VS, Rosen BK, Waicukauski JA (1990) On computing the sizes of detected delay faults. IEEE Trans Comput-Aided Des 9(3):299–312
22. Kajihara S, Taniguchi K, Miyase K, Pomeranz I, Reddy SM (2002) Test data compression using don't-care identification and statistical encoding. In: Proc. 11th IEEE Asian test symposium, pp 67–72
23. Kajihara S, Taniguchi K, Pomeranz I, Reddy SM (2002) Test data compression using don't-care identification and statistical encoding, In: Proc. first IEEE international workshop on electronic design, test and applications, pp 413–416
24. Kantipudi KR (2007) Minimizing N-detect tests for combinational circuits. Master's thesis, Auburn University, ECE Dept.
25. Kantipudi KR, Agrawal VD (2007) A reduced complexity algorithm for minimizing N-detect tests. In: Proc. 20th international conf. VLSI design, pp 492–497
26. Kavousianos X, Chakrabarty K (2011) Generation of compact stuck-at test sets targeting unmodeled defects. IEEE Trans Comput-Aided Des 30(5):787–791
27. Lavo DB, Larrabee T (2001) Making cause-effect cost effective: low-resolution fault dictionaries. In: Proc. international test conf., pp 278–286
28. Lin Y, Lu F, Cheng KT (2007) Multiple-fault diagnosis based on adaptive diagnostic test pattern generation. IEEE Trans Comput-Aided Des 26(5):932–942
29. Lin YC, Cheng KT (2006) Multiple-fault diagnosis based on single-fault activation and single-output observation. In: Proc. design, automation and test in Europe, pp 424–429
30. Majhi AK, Agrawal VD, Jacob J, Patnaik LM (2000) Line coverage of path delay faults. IEEE Trans VLSI Syst 8(5):610–614
31. McCluskey EJ, Tseng CW (2000) Stuck-fault tests vs. actual defects. In: Proc international test conf., pp 336–343
32. Mentor Graphics (2004) FastScan and FlexTest Reference Manual
33. Mentor Graphics (2009) DFTAdvisor Reference Manual
34. Pomeranz I, Reddy SM (1992) On the generation of small dictionaries for fault location. In: Proc. IEEE/ACM international conference on computer-aided design, pp 272–279
35. Pomeranz I, Reddy SM (2000) A diagnostic test generation procedure for synchronous sequential circuits based on test elimination by vector omission for synchronous sequential circuits. IEEE Trans. Comput-Aided Des 19(5):589–600
36. Pomeranz I, Reddy SM (2007) Diagnostic test generation based on subsets of faults. In: Proc. 12th IEEE European test symp., pp 151–158
37. Pomeranz I, Reddy SM, Venkataraman S (2007) z-diagnosis: a framework for diagnostic fault simulation and test generation utilizing subsets of outputs. IEEE Trans Comput-Aided Des 26(9):1700–1712
38. Pramanick AK, Reddy SM (1997) On the fault coverage of gate delay fault detecting tests. IEEE Trans Comput-Aided Des 16(1):78–94
39. Python https://www.python.org/download/releases/3.4.1/. Accessed July 1 2014
40. Ryan PG, Fuchs WK, Pomeranz I (1993) Fault dictionary compression and equivalence class computation for sequential circuits. In: Proc. IEEE/ACM international conference on computer-aided design, pp 508–511
41. Sandireddy RKKR (2005) Hierarchical fault collapsing for logic circuits. Master's thesis, Auburn University, ECE Dept.
42. Sandireddy RKKR, Agrawal VD (2005) Diagnostic and detection fault collapsing for multiple output circuits. In: Proc. design, automation and test in Europe, pp 1014–1019
43. Shukoor MA (2009) Fault detection and diagnostic test set minimization. Master's thesis, Auburn University, ECE Dept.
44. Shukoor MA, Agrawal VD (2009) A two phase approach for minimal diagnostic test set generation. In: Proc. European test symp., pp 115–120
45. Shukoor MA, Agrawal VD (2012) Diagnostic test set minimization and full-response fault dictionary. J Electron Test Theory Appl 28(2):177–187
46. Veneris A, Chang R, Abadir MS, Amiri M (2004) Fault equivalence and diagnostic test generation using ATPG. In: Proc. international symposium on circuits and systems, pp V–221–V–224
47. Venkataraman S, Hartanto I, Fuchs WK, Rudnick EM, Chakravarty S, Patel JH (1995) Rapid diagnostic fault simulation of stuck-at faults in sequential circuits using compact lists. In: Proc. design automation conference, pp 133–138
48. Wang Z, Marek-Sadowska M, Tsai K-H, Rajski J (2003) Multiple fault diagnosis using n-detection tests. In: Proc. 21st international conference on computer design, pp 198–201

49. Wang Z, Marek-Sadowska M, Tsai K-H, Rajski J (2006) Analysis and methodology for multiple-fault diagnosis. IEEE Trans Comput-Aided Des 25(3):558–575
50. Yu X, Amyeen ME, Venkataraman S, Guo R, Pomeranz I (2003) Concurrent execution of diagnostic fault simulation and equivalence identification during diagnostic test generation. In: Proc. 21st IEEE VLSI test symp., pp 351–356
51. Yu X, Wu J, Rudnick EM (2000) Diagnostic test generation for sequential circuits. In: Proc. international test conf., pp 225–234
52. Zhang Y (2012) Diagnostic test pattern generation and fault simulation for stuck-at and transition faults. PhD thesis, Auburn University, ECE Dept.
53. Zhang Y, Agrawal VD (2010) A diagnostic test generation system. In: Proc. international test conf., pp 360–368
54. Zhang Y, Agrawal VD (2010) An algorithm for diagnostic fault simulation. In: Proc. 11th IEEE Latin American test workshop, pp 1–5
55. Zhang Y, Agrawal VD (2011) Reduced complexity test generation algorithms for transition fault diagnosis. In: Proc. 29th IEEE international conf. computer design, pp 96–101

Yu Zhang graduated in 2012 with a PhD in Electrical and Computer Engineering from Auburn University, Alabama, USA. He worked with Dr. Vishwani Agrawal on electronic testing/diagnosis as a research/teaching assistant. He obtained his BS and ME degrees from Tianjin University, Tianjin, China in 2006 and 2008, respectively. He is currently a senior DFT (design for test) engineer at Broadcom Corporation, San Diego, USA.

Bei Zhang is a PhD student at Auburn University, Auburn, Alabama, USA. He received his BE and MS degrees in electrical engineering from Harbin Institute of Technology, China in 2008 and 2010, respectively. He was a graduate research and teaching assistant in the Department of Electrical and Computer Engineering at Auburn University since 2010. Between May 2013 and December 2013, he was an DFT intern at Broadcom Corporation, San Diego, California. His research interest includes VLSI testing, fault diagnosis, and testing of 3D stacked ICs.

Vishwani D. Agrawal is James J. Danaher Professor of Electrical and Computer Engineering at Auburn University, Alabama, USA. He has over forty years of industry and university experience, working at Bell Labs, Murray Hill, NJ, USA; Rutgers University, New Brunswick, NJ, USA; TRW, Redondo Beach, CA, USA; IIT, Delhi, India; EG&G, Albuquerque, NM, USA; and ATI, Champaign, IL, USA. His areas of expertise include VLSI testing, low-power design, and microwave antennas. He obtained his BE degree from the Indian Institute of Technology Roorkee, Roorkee, India, in 1964; ME degree from the Indian Institute of Science, Bangalore, India, in 1966; and PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign, in 1971. He has published over 350 papers, has coauthored five books and holds thirteen United States patents. He is the Editor-in-Chief of the *Journal of Electronic Testing: Theory and Applications*, and a past Editor-in-Chief (1985-87) of the *IEEE Design & Test of Computers* magazine. He was the Keynote Speaker at the 25th International Conference on VLSI Design, Hyderabad, India, 2012, Keynote Speaker at the Ninth Asian Test Symposium, Taipei, Taiwan, 2000, and an invited Plenary Speaker at the 1998 International Test Conference, Washington, D.C., USA. He served on the Board of Governors (1989-90) of the IEEE Computer Society, and in 1994 chaired the Fellow Selection Committee of that Society. He has received eight Best Paper Awards, two Lifetime Achievement Awards, and the 2014 James Monzel Award from the IEEE North Atlantic Test Workshop. Agrawal is a Fellow of the ACM, IEEE and IETE-India. He has served on the Advisory Boards of the ECE Departments of the University of Illinois at Urbana-Champaign, New Jersey Institute of Technology, and the City College of the City University of New York. See his website: http://www.eng.auburn.edu/~vagrawal.