

Diagnostic Test Generation and Fault Simulation Algorithms for Transition Faults*

Yu Zhang (Student Presenter) and Vishwani D. Agrawal

Auburn University, Department of Electrical and Computer Engineering, Auburn, AL 36849, USA

yz0009@auburn.edu, vagrawal@eng.auburn.edu

Abstract—To distinguish between a pair of transition faults, we need to find a test vector pair that produces different output responses for the two faults. By adding a few logic gates and one modeling flip-flop to the circuit under test (CUT), we create a diagnostic ATPG model usable by a conventional stuck-at fault test generator. Given a transition fault pair, this ATPG model either finds a distinguishing test or proves the faults to be equivalent. An efficient fault simulator is devised to find undistinguished fault pairs from a fault list for a certain test vector set. The number of fault pairs that needs to be targeted by the ATPG is greatly reduced after diagnostic fault simulation. A fault that is distinguished from all other faults is dropped from further simulation, thus making the complexity of diagnostic fault simulation similar to conventional fault simulation. We use a previously proposed diagnostic coverage (DC) metric to determine the distinguishability (diagnosability) of a test vector set. Experimental results show improved DC for benchmark circuits after applying the proposed diagnostic ATPG algorithms.

1 Introduction

A recent paper [5] describes a system for generating diagnostic tests using the single stuck-at fault model. Main ideas introduced there were a definition of diagnostic coverage and algorithms for diagnostic simulation and exclusive test generation. In that work emphasis was placed on using the existing tools that were originally designed for fault detection only.

The present work extends that capability to the diagnosis of transition faults, although a reader will find these extensions to be non-trivial. Once again we emphasize the use of existing tools and techniques that are freely used to build the new algorithms. The basic tools used are the simulation and test generation programs for detection of single stuck-at faults. Scan test

environment is assumed in which both launch off capture (LOC) and launch off scan (LOS) types of tests can be conducted. We borrow the diagnostic coverage (DC) metric from the previous paper [5]. The new modeling techniques and algorithms are more efficient than those published before [3]. Our implementation and results support the practicality of the presented approach.

2 Motivation for Transition Fault Diagnosis

The usefulness of the transition fault model stems from the fact that modern VLSI devices must be tested for performance. Transition faults are not perfect and in fact may not represent many of the actual defects. Their acceptability, like that of stuck-at faults, is due to several practical reasons. For example, their number grows only linearly with circuit size, they require two-pattern tests that are essential for detecting delay and other non-classical faults, and the scan methodology can be adapted to test them.

Why is the diagnosis of transition faults important? The same technology advances that give us lower cost and higher performance make it necessary that we diagnose delay defects. Presently, we must rely on ad-hoc measures like at-speed testing, N -detect vectors, etc. The present work is aimed at providing a similar diagnostic capability for transition faults as is available for stuck-at faults [5].

3 Modeling a Transition Fault

Because we wish to use the existing methods that are based on the logic description of the fault condition, we will use a synchronous model for the transition fault. Figure 1 shows a method of modeling a single slow-to-rise or slow-to-fall transition fault on a line xx' in the combinational logic of a synchronous sequential circuit. The shaded elements are inserted for modeling the fault and not part of the actual circuit.

*This research is supported in part by the National Science Foundation Grant CNS-0708962.

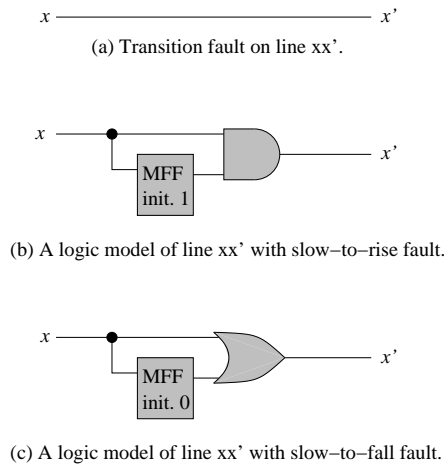


Figure 1. Modeling a faulty circuit with a single transition fault on line xx' .

The modeling flip-flop (MFF) is initialized to the value shown. Consider the slow-to-rise fault in Figure 1(b). Flip-flop initialization to 1 ensures that the output x' on the line will be the correct logic value on the first vector. Of the four two pattern sequences on x , 00, 01, 10 and 11, all except 01 will produce the correct output at x' . The sequence 01 at x will appear as 00 at x' , correctly representing a slow-to-rise transition fault on line xx' . Figure 1(c) shows a similar model for a slow-to-fall transition fault on line xx' .

4 An ATPG Model

An ATPG (automatic test pattern generation) model is a netlist of the circuit under test (CUT), modified to represent the target fault as a stuck-at fault. The modification amounts to insertion of a few logic elements for modeling only. For a transition fault, we construct the ATPG model as shown in Figure 2. The ATPG model of Figure 2(a) gives the conventional Boolean satisfiability formulation. Note that a 1 output from the EXOR gate cannot be obtained by a single vector. Because the modeling flip-flop MFF is initialized to 1, initially, $x' = x$. To produce a different output from the faulty circuit, the first vector must set $x = 0$ and then a second vector should set $x = 1$, besides sensitizing a path from x' to the primary output (PO).

The ATPG model of Figure 2(b) can be used in the same way [5]. Any test sequence for either s-a-0 or s-a-1 fault on y must produce different outputs from the fault-free and faulty circuits. The advantage of this model is that it can be simplified to use a single copy of the circuit. The analysis that leads to the

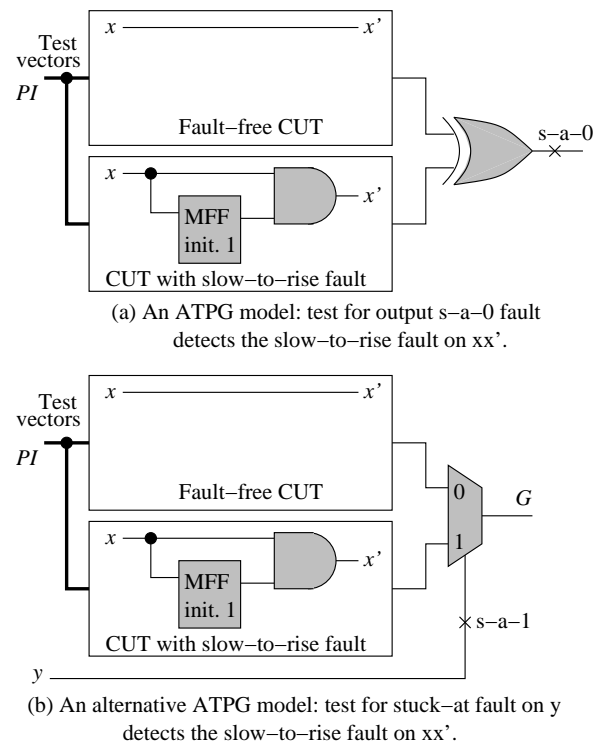


Figure 2. ATPG models in which a test for a stuck-at fault detects a slow-to-rise fault on line xx' in a circuit under test (CUT).

ATPG model of Figure 3 is the same as given in a recent paper [5]. There a single-copy ATPG model was obtained for finding an exclusive test for a pair of stuck-at faults. Thus, the fault-free CUT in Figures 2 (a) and (b) was replaced by the CUT containing one of the faults.

The main idea that allows us to collapse the two copies of the circuit in Figure 2(b) into a single copy is the realization that the two circuits are almost identical. The only difference is at the faulty line. It can be shown [5] that the multiplexer at PO can be moved to the fault site. The procedure is as follows: Suppose a transition fault is to be detected on a signal interconnect from x (source) to x' (destination). In a single copy of the circuit, the source signal x is made to fan out as two signals x_1 and x_2 . Fanout x_1 is left as fault-free signal. The other fanout x_2 is modified according to Figure 1 to produce the faulty value. These two signals x_1 and x_2 are applied to the two data inputs of a multiplexer whose output is x' , now feeding the destinations of the original x' , and control input is the new PI y . The target fault now is any stuck-at fault (s-a-0 or s-a-1) on y . Any test for this target must produce different values at fault-free x_1 and the faulty x_2 while propagating the value of x' to

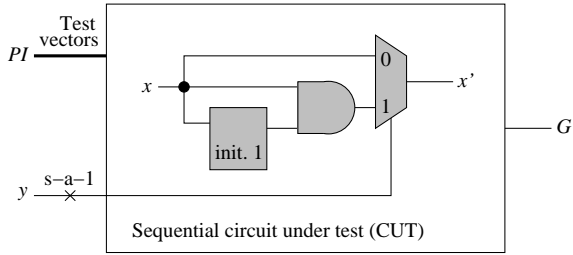
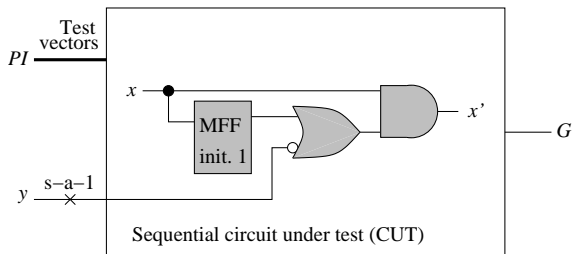
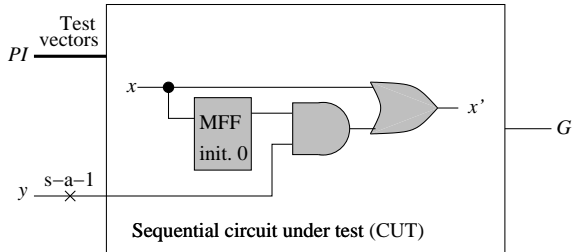


Figure 3. A single circuit copy ATPG model in which a test for a stuck-at fault on y detects the slow-to-rise fault on line xx' .



(a) Slow-to-rise transition fault on line xx' .



(b) Slow-to-fall transition fault on line xx' .

Figure 4. Simplified single circuit copy ATPG models in which a test for a stuck-at fault on y detects a transition fault on line xx' .

a PO, and hence must detect the fault modeled by $x2$. The resulting ATPG model for a slow-to-rise fault on xx' is shown in Figure 3.

Any test for y s-a-0 or for y s-a-1 in the ATPG model of Figure 3 will always contain two vectors. The model for a slow-to-fall transition fault is obtained by replacing the AND gate by an OR gate and changing the initialization of the flip-flop to 0, as shown in Figure 1(c). The gate and multiplexer combination can be further simplified to an equivalent ATPG model given in Figure 4, which shows the ATPG models for both slow-to-rise and slow-to-fall transition faults.

5 Combinational and Sequential Circuits

The preceding procedure of modeling a transition fault as a single stuck-at fault is valid for both combinational and scanned sequential circuits. For a combinational circuit under test (CUT), the modeling flip-flop (MFF) serves two purposes. First, it requires a two-vector test. Second, the initial state of the flip-flop makes it impossible to activate the fault effect at x' in the first vector. This model can be used to generate a two-vector test either by a sequential ATPG program or by a combinational ATPG program applied to a two time-frame expansion of the circuit.

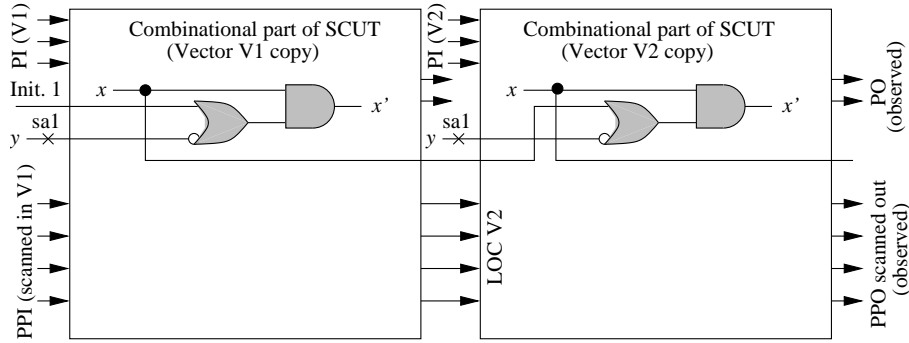
For a scanned sequential circuit under test (CUT) the ATPG models of the previous section will also generate two-vector tests. The vectors can be generated either by a scan ATPG program in the partial scan mode to accommodate the modeling flip-flop (MFF) or by a combinational ATPG program. The second vector would be generated either as a launch-off-capture (LOC) sequence or as a launch-off-shift (LOS) sequence. Figure 5 shows the two time-frame circuit for a combinational ATPG program.

6 Scan Test Generation

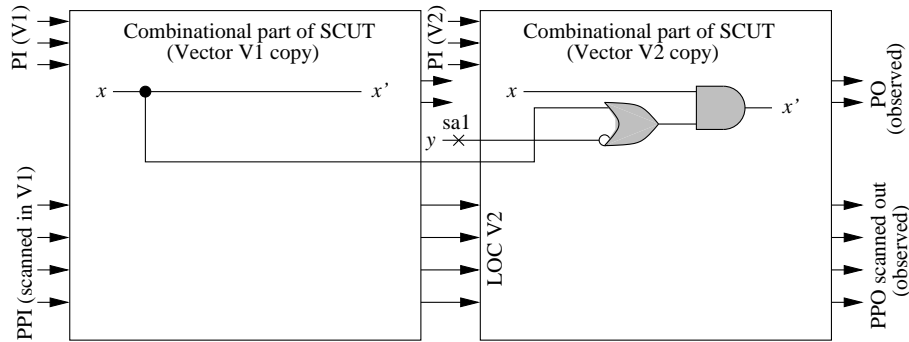
Consider a sequential circuit to be tested via scan. An ATPG tool like Mentor's Fastscan [4] will generate scan sequences for all stuck-at faults in the combinational logic of the circuit. Fastscan can also generate two-pattern scan sequences for all transition faults in the combinational logic. At the user's option, it generates tests for application in either LOC or LOS mode. Fastscan allows test generation in the partial scan mode as well provided the number of non-scan flip-flops is small, typically, less than eight. That capability is useful for the ATPG model of Figure 3 which requires a single non-scan flip-flop.

This ATPG model allows test generation for transition faults using the conventional stuck-at fault test generation tools. Fastscan, however, can directly generate tests as well as simulate them for transition faults. In our experiments, we use that capability of Fastscan. The ATPG model of Figure 3 will be used in later sections for generating exclusive tests for transition faults.

These models are especially useful when we generate tests using a combinational ATPG program. Both types of two-vector ($V1, V2$) tests, namely, LOC and LOS, can be generated. Figure 5(a) shows a combinational circuit for LOS test of a slow-to-rise fault. It



(a) Two time-frame combinational circuit for LOC test of slow-to-rise transition fault on line xx' .



(b) Simplified combinational ATPG circuit; a test for y s-a-1 is a LOC test for slow-to-rise fault on xx'

Figure 5. Two time-frame circuit for a LOC transition test for slow-to-rise fault on line xx' by a combinational ATPG program targeting the fault y s-a-1.

contains two copies of the combinational part of the sequential circuit. The fault is modeled using the construction of Figure 4(a). In the first time-frame the initial state, 1 (shown as *init. 1*), of the unscanned fault modeling flip-flop FF is applied through an extra primary input (PI) fixed at 1. All scan flip-flops (SFF) are stripped off and replaced by pseudo primary inputs (PPI) and pseudo primary outputs (PPO). Vector $V1$ consists of the normal PI and PPI. Vector $V2$ consists of the PI of the second time-frame where the PPI are the PPO of the first time-frame. All outputs of the second time frame are observable, PO directly and PPO through scanout. The circuit of Figure 5(a) has two faults. A closer examination, however, shows that it is impossible for the first y s-a-1 fault to cause any effect in the first time-frame due to the fixed “*init. 1*” input. Thus, the circuit can be simplified as shown in Figure 5(b) with a single stuck-at fault y s-a-1 for which any conventional combinational ATPG program can be used to obtain a test.

Figures 6(a) and (b) show two time-frame combinational circuit for a LOS test for a slow-to-rise tran-

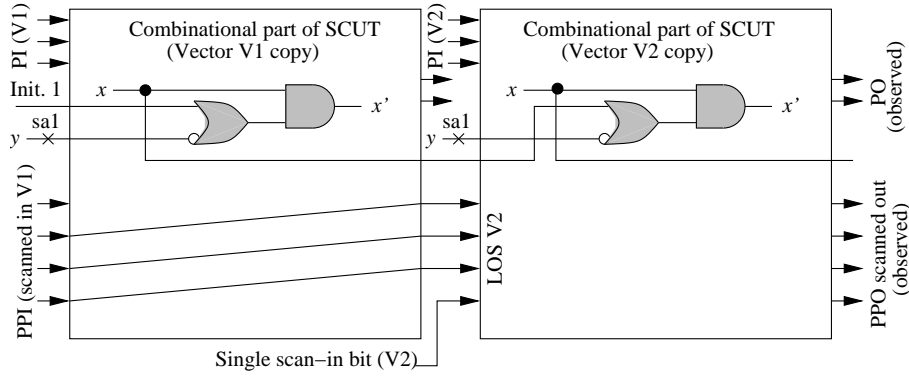
sition fault. The basic difference from the LOC model of Figures 5(a) and (b) is in the way the PPI bits are obtained in the second time-frame. For LOC test these bits are obtained by a one-bit shift of the PPI bits of $V1$.

Similar combinational circuit models for LOC and LOS tests can be obtained for a slow-to-fall transition fault by using the equivalent circuit of Figure 4(b).

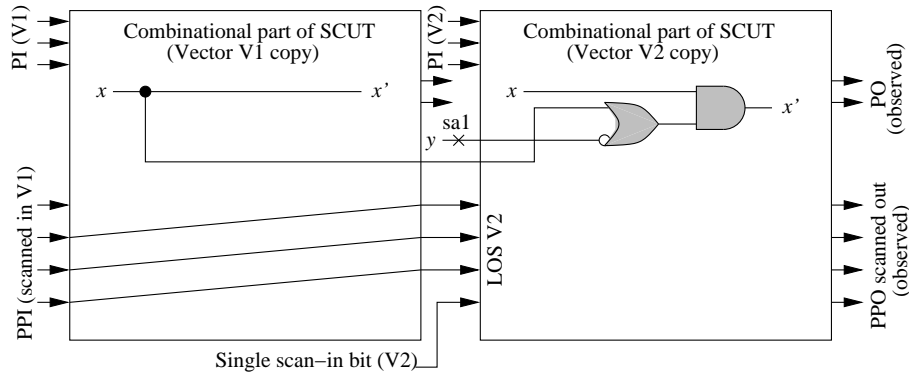
7 Diagnostic Test Generation

The main contribution of previous sections is modeling of a transition fault as a single stuck-at fault. The benefit of this model is that we can use the tools and techniques available for single stuck-at faults. We now illustrate the use of the following techniques discussed in a recent paper [5] for transition faults:

1. A diagnostic coverage (DC) that measures the ability to distinguish between any pair of faults.
2. Diagnostic fault simulator that determines DC for



(a) Two time-frame combinational circuit for LOS test of slow-to-rise transition fault on line xx' .



(b) Simplified combinational ATPG circuit; a test for $y\text{-}s\text{-}a\text{-}1$ is a LOS test for slow-to-rise fault on xx

Figure 6. Two time-frame circuit for a LOS transition test for slow-to-rise fault on line xx' by a combinational ATPG program targeting the fault $y\text{-}s\text{-}a\text{-}1$.

any given set of vectors and identifies undistinguished fault pairs. This diagnostic fault simulator internally uses any conventional single stuck-at fault simulator.

3. Exclusive test generator that derives an exclusive test for a fault pair such that the two faults in the pair can be distinguished from each other. If an exclusive test is found to be impossible then the two faults are equivalent and one of them can be removed from the fault set to further collapse it. This exclusive test generator internally uses a conventional single stuck-at fault test generator.
4. A complete diagnostic test generation system that first generates the conventional tests for fault detection coverage, determines the DC of those tests, and then generates more vectors if necessary to enhance DC.

The results of these procedures when applied to transition faults are shown in Table 1, which gives two types of coverages [5]. For a set of vectors we group faults such that all faults within a group are

not distinguishable from each other by those vectors, while each fault in a group is pair-wise distinguishable from all faults in every other group. This grouping is similar to equivalence collapsing except here grouping is conditional to the vectors. If we generate a new vector that detects a subset of faults in a group then that group is partitioned into two groups, one containing the detected subset and the other containing the rest. For multi-output circuit, the targeted group may be divided into more than 2 sub groups. Suppose, we have sufficient vectors to distinguish between every fault pair, then there will be as many groups as faults and every group will have just one fault. Prior to test generation all faults are in a single group we will call g_0 . As tests are generated, detected faults leave g_0 and start forming new groups, g_1, g_2, \dots, g_n , where n is the number of distinguishable fault groups. For perfect detection tests g_0 will be a null set and for perfect diagnostic tests, $n = N$, where N is the total number of faults. We define *diagnostic coverage, DC*, as

$$DC = \frac{\text{Number of detected fault groups}}{\text{Total number of faults}} = \frac{n}{N} \quad (1)$$

Table 1. Transition fault diagnostic test generation for ISCAS’89 benchmark circuits. Circuits have full scan and tests are generated for application in LOC mode.

Circuit	No. of faults	Detection test generation					Diagnostic test generation			
		Detection tests	FC %	DC %	Undiagnosed fault groups	Largest group	Exclusive tests	DC %	Undiagnosed fault groups	Largest group
s27	46	11	100.0	52.2	12	7	18	97.8	1	2
s298	482	44	79.9	62.4	62	5	34	70.1	39	4
s382	616	51	80.8	64.1	82	4	24	68.5	58	4
s1423	2364	102	92.9	79.3	280	5	106	84.1	182	5
s5378	6589	205	91.2	82.0	400	9	472	90.0	85	7
s9234	10416	377	92.8	75.8	1219	11	597	82.1	754	8
s13207	14600	480	89.1	70.0	1707	20	543	74.1	1392	11
s15850	17517	306	87.6	71.2	1961	9	486	74.3	1565	7
s35932	52988	75	99.0	88.3	3737	6	725	90.2	2867	4
s38417	47888	244	98.4	87.5	4090	9	1336	91.0	2883	8
s38584	56226	395	95.7	86.7	4042	8	1793	90.3	2440	7

Initially, without any tests, $DC = 0$, and when all faults are detected and pair-wise distinguished, $DC = 1$. Also, the numerator in equation 1 is the number of fault dictionary syndromes [2] and the reciprocal of DC is the *diagnostic resolution* (DR) [1]. The detection fault coverage (FC) is given by,

$$FC = \frac{\text{Number of detected faults}}{\text{Total number of faults}} = \frac{N - |g_0|}{N} \quad (2)$$

We used Fastscan [4] to generate fault detection tests for transition faults. Fastscan can generate transition fault tests for full-scan circuits in either of the two (LOC and LOS) modes. The results of Table 1 are for LOC mode only. The equivalent circuits of Figure 4 provide an alternative method. Here the target transition fault is represented as a single stuck-at fault. The modeling flip-flop MFF starts with a specified initial state and is not scanned. Thus, Fastscan generates a test for a single stuck-at fault y s-a-1 in the partial scan mode; all normal flip-flops of the circuit are scanned and the modeling flip-flop MFF is not scanned. All flip-flops including MFF are assumed to have the same clock. Because of the initial state of the unscanned MFF, the fault cannot be detected by the first vector, which serves as the initialization vector. The test essentially consists of two combinational vectors, or a scan-in sequence, followed by one clock in normal mode (LOC) or in scan mode (LOS), capture, and a scan-out sequence.

The second column of Table 1 lists the number of transition faults. Faults on same fanout free interconnect and the input and output of a not gate are collapsed [4]. Also some of the redundant transition faults are identified during ATPG and they are re-

moved. The third column lists the number of LOC tests. Note that Fastscan performs test pattern compaction. Since in this work our focus is on the ATPG algorithm, we did not perform compaction on diagnostic test patterns. Each test consists of a scan-in, capture and scan-out sequence. The detection fault coverage (FC) of transition faults is given in column 4. Reasons of less than 100% FC are (a) aborted ATPG, (b) LOS mode not used, and (c) redundancy or untestability not identified. Because Fastscan for transition faults operates in sequential mode it often fails to identify redundancies. In our ongoing work we will use the combinational models of Figures 5 and 6 with pure combinational ATPG to improve the fault efficiency. Base on observations of several small IS-CAS’89 circuits, most aborted pairs are actually functionally equivalent. If all equivalencies are identified, similar to fault efficiency, “diagnostic efficiency” would be much higher than diagnostic coverage. This needs further investigation.

Column 5 of Table 1 gives the diagnostic coverage (DC) obtained from diagnostic fault simulation [5], which divides faults into groups. Group g_0 contains undetected faults. Groups with more than one fault contain the faults that are mutually undistinguished (or undiagnosed). Thus, circuit s27 has 12 such groups and the largest of those groups has 7 faults (see columns 6 and 7). Similarly, s5378 has 400 multi-fault undiagnosed groups, largest one containing 9 faults.

The purpose of diagnostic test generation is to derive exclusive tests that will provide pair-wise diagnosis of faults within groups. This is done by modeling a pair of transition faults as two stuck-at faults using the technique of Figure 4 and then using a single

