Impact of Multiple-Detect Test Patterns on Product Quality

Brady Benware, Chris Schuermyer,
Sreenevasan Ranganathan, Robert Madge,
Prabhu Krishnamurthy

LSI Logic Corporation
Gresham, OR 97030

Nagesh Tamarapalli, Kun-Han Tsai,
Janusz Rajski

Mentor Graphics Corporation
8005 S.W. Boeckman Road
Wilsonville, OR 97070, USA

Abstract

This paper presents the impact of multiple-detect test patterns on outgoing product quality. It introduces an ATPG tool that generates multiple-detect test patterns while maximizing the coverage of node-to-node bridging defects. Volume data obtained by testing a production ASIC with these new multiple-detect patterns shows increased defect screening capability and very good agreement with the bridging coverage estimated by the ATPG tool.

1. Introduction

One of the key objectives of manufacturing test is to ensure high quality of shipped parts while managing the cost of test. Scan–based DFT methodology, combined with ATPG tools, automate the generation of test patterns with very high fault coverage. The advantage of a structure-based ATPG tool is its high efficiency and effectiveness in generating a test set by targeting different fault models, such as stuck-at, transition, path delay, and I_{DDQ}. DFT tools assess the quality of test patterns by reporting the fault coverage of the target fault models. However, real defects may not always be detected by test patterns generated for the targeted fault model.

The stuck-at fault model has been used in DFT since the very beginning and, while showing some limitations and imperfections, it has demonstrated its robustness and adaptability. Even though the stuck-at fault model may not always model behavior of a faulty circuit it serves very well as a target, i.e. a test set developed to test stuck-at faults will also cover many other defects that do not behave as stuck-at faults.

Good understanding of bridging defects is at the center of explanation of the effectiveness of the stuck-at fault model. It also provides the key clues to its enhancements. In an experimental study of bridging faults in a state of the art microprocessor design [1] it has been observed that approximately 80% of all bridges occur between a node and Vcc or Vss, and 20% involve non-supply nodes. Global signals were involved in 70% of these defects and leaf-level signals contributed only 30%. In another experimental evaluation of scan tests for bridging defects [2] it was concluded that bridges with power rails contributed between 60% to 90% of all bridging defects.

It is clear that a test that detect a stuck-at fault on a node will detect a low resistive bridging defect with the supply lines. This is exactly the behavior of a node stuck-at-0 or stuck-at-1. However, the detection of node-to-node bridging defects is not guaranteed. If a stuck-at fault on a node is detected once, the probability of detecting a static bridging fault with another un-correlated node that has signal probability 50% is also 50% [3]. If the stuck-at fault is detected twice, the estimated probability of detecting the bridging fault with another node acting as an aggressor is 75%. Signal correlation may reduce the coverage of node-to-node bridging faults. It was observed [1] that a test set with greater than 95% stuck-at fault coverage produced only 33% coverage of node-to-node bridging faults. Most likely the disappointing coverage was an artifact of signal correlation. Typically a test set created by conventional ATPG aiming at single detection may have up to 6% of faults detected only once and up to 10% of faults detected only once or twice. This may result in inadequate coverage of node-to-node bridging defects.

In general, there are two directions to overcome the limitation and improve the test quality. One direction is to enhance the fault model by describing the defect behavior and presenting it in a suitable form to the ATPG tool. In this case the fault model is more precise and complex and the fault list is longer. The advanced fault models, like bridging faults and cross-talk effects, use physical layout information to compile the fault lists. A complete example of this approach is demonstrated in [2]. Here the possible bridges are identified by analysis of layout using weighted critical area and their behavior is modeled by different types of faults and a special netlist. The experimental results from the project show that

0-7803-8106-8/03 $17.00 Copyright 2003 IEEE

patterns created using this approach detected unique defective parts that were not detected by a high-coverage stuck-at test set. There are clear benefits of this approach in improving quality of test and reducing reliance on functional patterns.

This approach, however, requires substantial infrastructure and specialized tools. Even with the extensive infrastructure in place it might still be difficult to target all the bridging defects. In [2] only 10% of the top 400K bridges were targeted achieving approximately 27% coverage of the node-to-node bridges. In addition, test generation cannot be performed before physical design is done. This process dependency causes extra delay before the design taped out. Moreover, the complexity of ATPG algorithm for those advanced fault models might be too high to be practical for multi-million gate designs. Furthermore, the exact mechanism of various defects is very often unknown, which makes the modeling itself a challenging task. In the worst case the ATPG tool may have very precise but not robust enough targets to work with.

The alternative approach is to utilize the conventional fault models, such as stuck-at faults, and apply the same ATPG algorithm to generate more patterns that increase the probability of detecting non-modeled defects without using layout information. There are several proposed methods that target stuck-at faults multiple times to improve quality of scan test patterns [3, 4, 5, 6].

In one of the first experimental studies that used multiple-detect patterns to maximize coverage of node-to-node bridging defects, the patterns also ensured propagation to all primary outputs [7]. The objective was to provide very high coverage of defects as well as very good diagnostic resolution. In this study 2.26% of failing devices passed a complete suck-at fault test set. The diagnostic process identified that 79.2% of defects were consistent with stuck-at behavior, 6.9% with transition faults, and 13.9% with neither speed related nor stuck-at faults.

An important set of guidelines for multiple-detect ATPG was defined in Random Excitation and Deterministic Observation (REDO) scheme [3]. In this scheme, in order to reduce the overall defective part level for a device, observability of each site was increased by targeting stuck-at faults multiple times. The ATPG algorithm was upgraded to target faults that are located on the least observed sites. The ATPG algorithm used random decision order in fault targeting and fault simulation was modified to drop faults from the list only when they were detected a specified number of times. Experimental results confirmed that the defect-oriented patterns provided better screening than the traditional patterns.

In this paper we present a multiple-detect ATPG tool developed specifically to maximize the detection of bridging defects. We also introduce a new measure of quality of test aimed at capturing node-to-node bridging defects. The ATPG tool aims at maximizing the coverage of bridging faults within a budget of test patterns. The experimental results obtained on a production ASIC shows a significant improvement in quality of scan test and very good correlation with the bridging coverage estimates.

2. Metrics for multiple-detect patterns

The analysis considers two objectives. The first objective is to target the non-modeled defect whose detection depends on the values of more than one circuit node (e.g. bridging defect). In this case, the additional pattern to detect the same fault with different node assignments will increase the chance of detecting the defect regardless of the observation point.

The second objective is to target the non-modeled defect, which can only be seen on some observation points. In this case, ATPG should try to propagate the fault to more locations if possible. So the additional patterns generated beyond single detect set increase the number of fault observation points, hence increase the chance of detecting this type of defect.

Two metrics, *Bridging Coverage Estimate* (*BCE*) and *Fault Observation Coverage* (*FOC*), are introduced later in this section to measure the quality of a multiple-detect test set. The purpose of BCE is to measure the ability of a test set to detect bridging defects. On the other hand *FOC* is used to determine the overall utilization of observation points for a given test set. Larger FOC indicates that the targeted faults propagate to more observation points.

2.1 Metric for stable bridging defects

In this paper we focus on low resistive bridging defects. Figure 1 depicts several types of bridging defects between wires w and k. If wire w is bridged to power or ground, the defect is modeled by a stuck-at fault. In the case of a bridging defect resulting in signal k dominating w, $w_D=k$, the defect is detected if the stuck-at fault on w is detected and, k and w have opposite values. If the bridging defect behaves as an OR-type model, $w_{OR}=OR(w,k)$, the detection condition is to detect stuck-at-1 fault on wire w (denoted as $w_{@1}$) while k is set to 1.

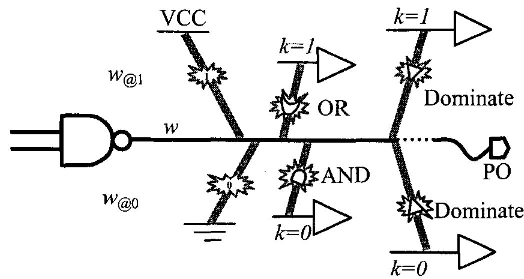


Figure 1 Bridging defect types

If the bridging defect behaves as an AND-type model, w_{AND}=AND(w,k), the detection condition is to detect stuck-at-0 fault on wire w (denoted as $w_{@0}$) while k is set to 0. The detection conditions of w bridging to k by propagating the stuck-at fault effect through the wire w are summarized in table 1. The cells in table 1 with a highlighted number indicate the cases that the corresponding type of bridging defect is detected.

According to the analysis, for the AND/OR type bridging defect, the probability of detecting the defect by stuck-at fault model depends on the chance of setting the right value on the involved wire. For example, lets assume that pattern t detects $w_{@1}$ and k is independent to the detection of $w_{@1}$. The probability that t detects w_{OR} is equal to the probability of k=1, which is assumed to be 0.5. If $w_{@1}$ is detected four times, the chance to detect the OR type bridge defect on w is (1- $(1\text{-}0.5)^4$), or 93.75%.

	w	k	OR	AND	D	V_{cc}	V_{ss}
$w_{@1}$	0	0	0	0	0	1	0
$w_{@1}$	0	1	1	0	1	1	0
$w_{@0}$	1	0	1	0	0	1	0
$w_{@0}$	1	1	1	1	1	1	0

Table 1 Bridging detect types and the detection condition

Based on the above analysis, the following metric is introduced to estimate the bridge fault coverage.

Definition: Given a test set T and target fault list F, the Bridging Coverage Estimate (BCE) is calculated as follows:

$$BCE = \sum_{i=1}^{n} \frac{f_i}{|F|} \cdot (1 - 2^{-i})$$

where f_i is the number of stuck-at faults detected i times by T, and $|F|$ is the total number of stuck-at faults in the target fault list F. n is the maximum number of detections that a fault can be detected by T. In practice, n is reduced to the maximum number of detections that the ATPG tool keeps track of. Once a fault is detected n times it is dropped from the target fault list. This treatment of n can lead to error in the calculated BCE. For example, if n is limited to 5, the upper bound of the BCE is 96.875%. However, n limited to 10 yields an upper bound of 99.9%, which in most cases is accurate enough to judge the quality of the test set. For consistency, the stuck-at fault coverage (SAF) is defined here as:

$$SAF = \sum_{i=1}^{n} \frac{f_i}{|F|}$$

The argument in the summation is simply the fraction of faults detected i times and is denoted in this paper as SAF_i. Similarly, BCE_i is used to denote the argument of the summation for BCE.

2.2 Metric for observation point sensitive defect

It is noticed that some bridging defects can be detected only at certain observation points. Figure 2 shows an example of such a bridging defect. The example assumes that the defect under consideration is an AND type bridging defect between nodes w and k. The bridging defect is detected if w stuck-at-0 fault is detected while k is set to 0. If w stuck-at-0 is observed at PO_2, the node k must be set to 1 to satisfy the propagation condition. Thus the bridging defect cannot be excited. The bridging defect can only be detected if the w stuck-at-0 fault propagates to PO_1. Propagation of faults to multiple outputs helps in de-

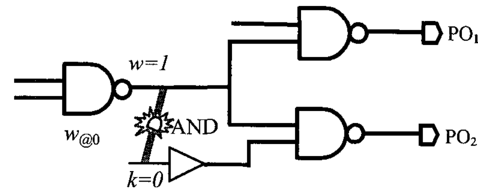


Figure 2 Example of observation point sensitive bridging defect

correlating the target node and the aggressor node. It also identifies cases where some faults are observable on very small number of observation points. Forcing the faults to propagate to different outputs creates different assignments and helps expose the bridging defects.

Since the ATPG tool does not consider the layout information, the best way to increase the probability of detecting bridging defects is by propagating the stuck-at faults to increased number of observation points. The following metric is proposed to measure the utilization of observation points by a test set.

Definition: Given a test set *T*, the Fault Observation Coverage (*FOC*) is defined as follows:

$$FOC = (\sum_{f \in F} op(f) \,/\, op_{max}(f)) \,/\, |F|$$

where $op(f)$ is the number of observation points utilized by T to observe a fault f, and $op_{max}(f)$ is the maximum number of possible observation points for fault f.

Example: Consider a circuit containing three faults: f_1, f_2, and f_3, and each fault has five possible observation points. Suppose that faults f_1, f_2, and f_3 are detected at 3, 2, and 4 observation points respectively. The corresponding *FOC* is calculated as below:

$$FOC = (3/5 + 2/5 + 4/5) / 3 = 60\ \%$$

The ATPG tool attempts to propagate the faults to different observation points so that *FOC* can be maximized. Instead of tracking observation points for all faults, the ATPG tool uses random decision order to achieve the same objective. In this approach, when a fault propagates through multiple fanout branches, the ATPG tool selects a fanout branch randomly. In addition, during fault excitation and assignment justification, random decision is made when multiple choices exist.

By using random decision order during test pattern generation, the probability that different patterns utilize the same observation point is significantly reduced. This can be validated by the following experiment that compares *FOC* for multiple detection test sets with and without random decision order.

Validation Experiment: The test generation is done in the following manner with and without random decision order:

1. Perform single-detect ATPG for stuck-at faults to generate test set T1.
2. Perform multiple-detect fault simulation (n=5) for T1 and extract fault list, F1, which contain the faults detected only once.
3. Perform ATPG four times targeting F1 and generate four test sets T2 ,..., T5.

The experimental results for five industrial circuits ranging from 50K to 1.5M gates are shown in table 2. The columns labeled 1 to 5 show the average number of observation points utilized by test sets T1 to T5. The last column, labeled obmax shows the maximum number of reachable observation points averaged across all faults. The maximum number of observation points for each fault is calculated based on the structural analysis of reachable observation points. Each circuit contains two rows of data that report the average number of observation points utilized with and without random decision order.

	i	1	2	3	4	5	op_{max}
C1	rand.	4.61	5.13	7.34	8.71	9.77	16.90
	w/o rand.	3.86	3.99	5.65	6.43	7.01	
C2	rand.	4.56	5.41	7.41	9.04	10.39	26.81
	w/o rand.	3.45	3.49	4.48	4.97	5.23	
C3	rand.	6.27	6.61	8.91	10.20	10.78	17.02
	w/o rand.	4.37	4.44	5.78	6.11	6.67	
C4	rand.	4.94	5.55	8.68	10.68	12.02	19.16
	w/o rand.	3.71	3.74	4.80	5.33	5.51	
C5	rand.	10.44	10.94	11.26	11.55	11.86	16.88
	w/o rand.	7.07	7.21	7.37	7.46	7.58	

Table 2 FOC validation experiment result

Figure 3 compares the average *FOC* of these five industrial circuits for five-detect test sets with and without random decision order. The chart illustrates that random decision order significantly increases the number of utilized observation points. Note that for ATPG test sets generated without random decision order, the increase in FOC is due to the different random fill employed in different ATPG runs.

The *BCE* and *FOC* metrics can be used to measure the quality of a test set beyond the single stuck-at fault coverage. In this experiment, for example, all test sets have identical stuck-at fault coverage but different effectiveness in detecting bridging defects.

The proposed algorithm uses the *BCE* metric to guide the ATPG tool to generate the multiple-detect test

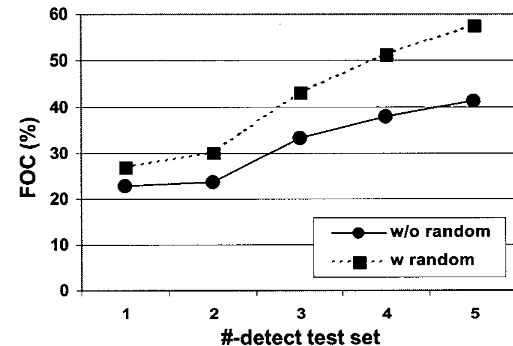


Figure 3 FOC comparison between ATPG test sets with and without random decision

patterns. *FOC* is not used directly during ATPG to avoid the complexity of tracing all observation points. Instead, the random decision order is used during ATPG to improve the *FOC* metric implicitly.

When the multiple-detect test set exceeds the capacity of the tester memory, these metrics can be used to guide the truncation of the test set.

3. Multiple detect pattern generation methodology

The methodology utilized to generate multiple-detect patterns is illustrated in figure 4. The initial step is to determine the list of faults, TF_{MD}. These are the faults detected by the single-detect pattern set T_1. This is done by performing single-detect fault simulation with T_1 for all faults in the circuit. Multiple-detect patterns are generated only for faults that are detected by the single-detect test set. This ensures that the multiple detect pattern set does not detect any additional stuck-at faults that were not detected by the original single detect pattern and enables proper comparison of quality levels between the single-detect and multiple-detect pattern set.

1. Perform *single-detect* fault simulation with *single-detect* pattern set T_1 for all faults
2. Save all faults detected by single-detect fault simulation with pattern set T_1 (TF_{MD})
3. Set the number of detections N
4. For $K = 1$ to $(N-1)$
   - Perform *multiple-detect* fault simulation with pattern sets T_1 to T_K for TF_{MD} faults
   - Save faults detected K times (F_K)
   - Target faults F_K and perform *single-detect* ATPG to increase the number of detections by one
   - Save the patterns to $T_{(K+1)}$
5. Perform *multiple-detect* fault simulation with pattern sets T_1 to T_N for all faults to obtain multiple-detect fault coverage profile

Figure 4 Multiple-detect ATPG methodology

Once the list of faults detected by the single-detect pattern set is established, the multiple-detect pattern set is generated in an iterative manner. In each iteration K, multiple-detect fault simulation is performed with pattern sets that are generated so far, T_1 to T_K, for F_1 faults. Faults that are detected K times are saved and single-detect ATPG is performed to increase the number of detections of these faults by one, from K to $(K+1)$.

Finally, a multiple-detect fault simulation is performed with all pattern sets T_1 to T_N for all faults in order to determine the multiple-detect fault coverage profile. This profile can be used to determine the *BCE* metric and assess how well it correlates with silicon results. The final fault simulation is also useful in determining the list of faults detected by the complete multiple-detect pattern set. This list of detected faults must match with the faults detected by the original single-detect ATPG pattern set T_1. Note that one of the advantages of performing the multiple-detect ATPG in the manner described here is that the effectiveness of each additional detection can be easily gauged since the pattern set for each detection is saved separately.

Table 3 shows the fault coverage statistics for the multiple detect pattern set that was generated following this methodology for the ASIC used in this work. The statistics presented for each additional pattern T_2 to T_5 are the cumulative result of that pattern and the previous ones. The faults detected six to nine times are not shown in the table to save space, but were considered in calculating the total SAF and BCE in the final columns of the table.

The first observation from this data is that the original production single detect pattern T_1 has a stuck-at fault coverage of 96.85% and a BCE of only 90.66%. It is also possible to see that the biggest contributor to the low BCE coverage is due to the 7.73% of the stuck-at faults that are only detected once. For this reason, these are the faults that are targeted in the generation of T_2 as described above. Notice that with the combined test set of T_1 and T_2, the number of faults detected only once drops to 0.01% and the corresponding BCE increases to 94.25%. Overall, the full pattern set T_1 to T_5 detects 99.99% of the targeted faults five or more times and results in an overall increase in BCE of 5.86%. Furthermore, the stuck-at fault coverage has essentially remained constant ensuring that any additional devices failed for patterns T_2 to T_5 is due only to the multiple detections of faults.

Also shown in table 3, below the test pattern name is the cumulative pattern depth. Notice that the multiple detect test set has about 4x the number of partitions of the single detect pattern T_1. In general, it may not be possible to store all the multiple-detect patterns on the tester due to tester memory limitations. In such cases, the pattern set for each additional detection can be ordered using *BCE* and truncated appropriately.

4. Experimental setup

To determine the effectiveness of multiple detect pattern sets, volume data was collected on one production ASIC running in LSI Logic's 0.18μm Al process. The ASIC has five metal layers and has a

Cumulative Test Set	i	1	2	3	4	5	10+	Total SAF	Total BCE
	$1-2^{-i}$	0.500	0.750	0.875	0.938	0.969	0.999	SAF	BCE
T_1 (4236)	$SAF_i(\%)$	7.73%	5.87%	3.90%	3.04%	2.22%	67.99%	96.85%	90.66%
	$BCE_i(\%)$	3.87%	4.41%	3.41%	2.85%	2.15%	67.92%		
	$\Delta_i(\%)$	3.87%	1.47%	0.49%	0.19%	0.07%	0.07%		
T_2 (7492)	$SAF_i(\%)$	0.01%	6.84%	3.54%	4.00%	2.41%	73.61%	96.86%	94.25%
	$BCE_i(\%)$	0.00%	5.13%	3.10%	3.75%	2.34%	73.54%		
	$\Delta_i(\%)$	0.00%	1.71%	0.44%	0.25%	0.08%	0.07%		
T_3 (10719)	$SAF_i(\%)$	0.00%	0.01%	6.50%	2.97%	2.52%	77.34%	96.86%	95.63%
	$BCE_i(\%)$	0.00%	0.00%	5.68%	2.78%	2.44%	77.27%		
	$\Delta_i(\%)$	0.00%	0.00%	0.81%	0.19%	0.08%	0.08%		
T_4 (13969)	$SAF_i(\%)$	0.00%	0.00%	0.01%	6.38%	2.65%	79.96%	96.86%	96.23%
	$BCE_i(\%)$	0.00%	0.00%	0.01%	5.99%	2.56%	79.89%		
	$\Delta_i(\%)$	0.00%	0.00%	0.00%	0.40%	0.08%	0.08%		
T_5 (17190)	$SAF_i(\%)$	0.00%	0.00%	0.00%	0.01%	6.29%	83.05%	96.86%	96.51%
	$BCE_i(\%)$	0.00%	0.00%	0.00%	0.01%	6.09%	82.97%		
	$\Delta_i(\%)$	0.00%	0.00%	0.00%	0.00%	0.20%	0.08%		
					Overall Improvement			0.002%	5.856%

Table 3 SAF and BCE statistics for the full multiple detect pattern set T1-T5

total of 944,350 stuck-at faults. The multiple-detect pattern set was generated as described above, however, due to tester memory limitations, only the first 1000 scan partitions from each pattern T_2 to T_5 were used. The coverage for the truncated multiple detect pattern set is summarized in table 4. Patterns T_2 to T_5 were all tested at a frequency of < 1MHz and integrated into the test program just after the production scan pattern T_1. The only tests that precede the production scan test are continuity, power shorts, gross I_{DDQ}, and a process monitor test. Devices that fail any one of these tests, including the production scan test would fail immediately and not be subjected to the multiple-detect pattern set. Devices that did pass all these tests were then tested to patterns T_2 to T_5. During the multiple-detect test the pass/fail result was recorded for each section of the multiple-detect pattern set. All devices, irrespective of their pass/fail status in the multiple-detect pattern set, were subjected to the rest of the test program. The intent was to determine the relative

Cumulative Test Set	SAF	BCE
T_1 (4236)	96.85%	90.66%
T_2 (5237)	96.86%	93.50%
T_3 (6238)	96.86%	94.70%
T_4 (7239)	96.86%	95.31%
T_5 (8240)	96.86%	95.66%
Overall Improvement	0.001%	5.003%

Table 4 SAF and BCE statistics for truncated multiple detect pattern set T1-T5

effectiveness of each section of multiple-detect pattern set as well as determine what other tests if any were screening these defective devices.

Of particular interest is the overlap between multiple-detect failures and I_{DDQ} outlier die. Effective I_{DDQ} outlier screening through off-tester Statistical Post-Processing™ (SPP) has been shown to significantly reduce Early Failure Rate (EFR) failures through burn-in [8]. It has also been shown that the outlier screening with these methods reduces the customer DPM by screening potential test escapes. In particular, I_{DDQ} outlier screening should detect a significant amount of bridging defects because I_{DDQ} is providing additional detections of faults.

5. Experimental results

Data was collected on many production lots and represents a total of >200,000 die tested. Figure 5 shows the individual and cumulative fallout obtained for each section of the multiple-detect pattern set. These failures represent die that pass the regular stuck-at scan single-detect pattern T_1, but fail one or more sections of the multiple-detect pattern set. The bar chart shows that roughly an equal number of devices fail for each section, which is consistent with the roughly equal fault coverage of each section. Also shown is the cumulative fallout, which is the total number of unique failures after each additional section is applied. In this case, a total of 70 unique die were failed by patterns T_2 to T_5. Even though the application of each successive section results in the detection of fewer additional devices, as can be seen from the results, even the last section of the multiple-

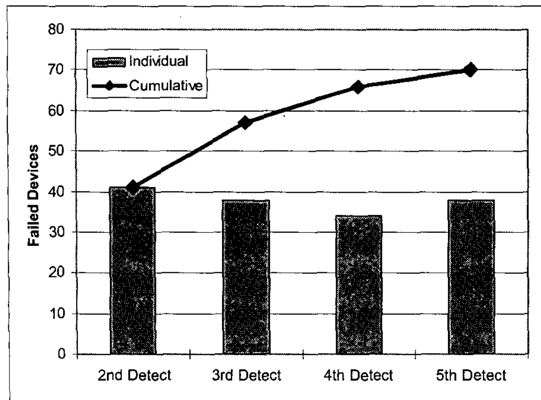


Figure 5 Individual and cumulative fallout observed for the multiple detect patterns on ASIC1

detect pattern set detects a significant number of failing devices.

As described above, the devices failing the multiple-detect pattern set were subjected to all regular production tests to determine if these faulty devices were being screened by other tests. The pie chart in figure 6 shows the breakout of test results for these multiple-detect failures. The majority of the devices go on to fail the memory test. Although it has not explicitly been determined, it is believed that the memory failures are independent defects due to the very low amount of common logic targeted by the two tests. In addition, it is possible to see that of the devices that don't fail the memory test, about 40% failed the SPP I_{DDQ} screening methods presented in [8].

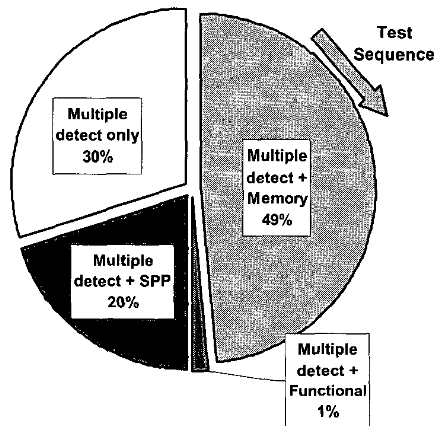


Figure 6 The additional test results of the devices that failed the multiple-detect test set

From the previous data, it is clear that patterns that only detect previously detected faults do provide additional benefit not realized by single detect patterns alone. However, it is also important to understand how well the metrics used to measure pattern quality can predict actual test quality. Figure 7 compares the relative cumulative fallout to the relative BCE increase of each additional pattern. For example, pattern T2 fails 40 out of the 70 total failures to the full pattern set or 57.1%. Similarly, as can be seen in table 4. pattern T2 accounts for 2.84% out of the 5.00% total BCE increase or 56.7%. The data shows strong correlation between the relative cumulative fallout and relative *BCE* increase and confirms that *BCE* can be used during multiple-detect ATPG to measure the quality of the test set.

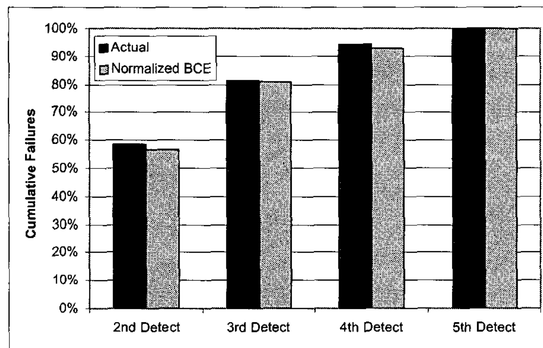
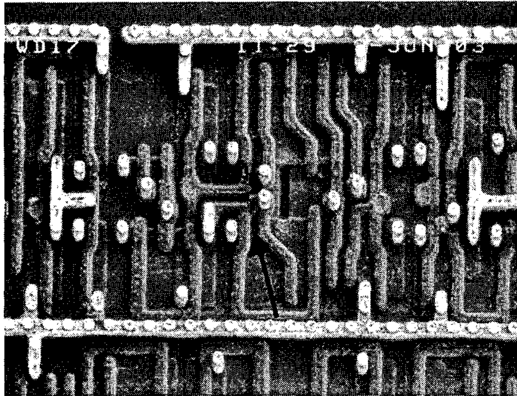


Figure 7 Comparison of normalized BCE and cumulative fallout percentage for multiple detect test sets

6. Failure analysis

At the time of this writing, failure analysis has only been performed on one device that was a multiple detect only failure. This device failed patterns T_2, T_3 and T_5, but passed T_4 and the original single stuck-at test T_1. Datalogs were collected on the ATE and scan-diagnosis was performed using the ATPG tool. Diagnosis was performed on each failing pattern individually and each diagnosis returned the same result, a stuck-at zero on one net. Through electrical and visual verification, the failure analysis determined that a poly-silicon particle was shorting the gate of the diagnosed net to the gate of another signal line as the pictures show in figure 8 and confirms the existence of a bridging defect. To be certain that this defect was screened due to multiple detects and not additional stuck-at fault coverage, the identified stuck-at fault was simulated with the ATPG tool and revealed that this fault was detected exactly once by each applied pattern T1 to T5. The assigned probability of detecting a bridging defect is 50% per detect and is fairly consistent with the 3/5 single detect patterns that failed.



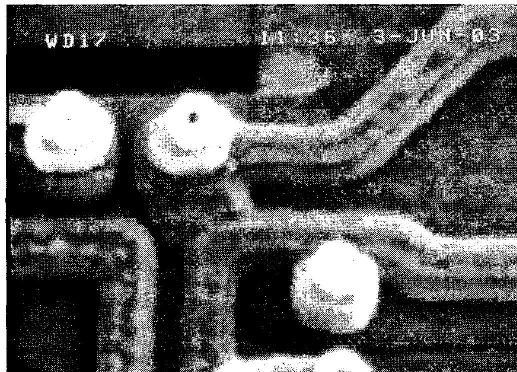


Figure 8 **SEM image of a poly-silicon bridging defect screened with the multiple-detect pattern set (Top). Close up of defect (Bottom).**

7. Estimation of defect level and bridging defect occurrence

While the results of the previous section demonstrate the ability of multiple detect patterns to screen unique defects, the true impact of these patterns should be gauged by their ability to reduce the defect level. The defect level (typically measured in defective parts per million or DPM) is the fraction of defective parts in a population of parts that are thought to be defect-free and can be expressed as [9]

$$D_L = 1 - \frac{Y}{Y_a(\Omega)} \tag{1}$$

where Y is the true yield and $Y_a(\Omega)$ is the apparent or measured yield given a defect coverage of Ω. Note that the true yield Y is equal to the apparent yield if the defect coverage is 100%. Therefore, with an appropriate yield model, the difficulty in determining the defect level lies in determining the defect coverage given a test set *T*. The task of estimating defect coverage from a know stuck-at fault coverage has been thoroughly explored in [9-12]. However, in all of these approaches, a stuck-at fault coverage of 100% translates into a defect coverage of 100%, which from the data presented in the previous section shows that stuck-at coverage alone cannot yield 100% defect coverage. Given the data presented in this paper, it is not possible to develop a full defect level model to replace the existing approaches. Therefore, the focus of the following is to present the expected change in the defect level given a known change in *BCE* coverage for a test set.

The first step in developing a model for defect level is to choose an appropriate yield model. The negative binomial yield model has been chosen in this work due to its good agreement with data and its ability to approximate most other yield models. The apparent and true yield are given by

$$Y_a(\Omega) = \left(1 + \frac{A \cdot D_0 \cdot \Omega}{\alpha}\right)^{-\alpha} \tag{2}$$

and,

$$Y = Y_a(1) = \left(1 + \frac{A \cdot D_0}{\alpha}\right)^{-\alpha} \tag{3}$$

where A is the chip area, D_0 is the true defect density and α is a parameter that describes the defect clustering on a wafer. Note, at the limit of $\alpha=\infty$ the yield equation reduces to the Poisson yield equation and represents a perfectly random distribution of defects. Reworking (3),

$$\frac{A \cdot D_0}{\alpha} = \left(Y^{-1/\alpha} - 1\right) \tag{4}$$

and substituting into (2), the apparent yield can be expressed as a function of true yield and defect coverage:

$$Y_a = \left[1 + \left(Y^{-1/\alpha} - 1\right) \cdot \Omega\right]^{-\alpha} \tag{5}$$

However, what is desired is to express the change in apparent yield as a function of the change in defect coverage. Therefore, the derivative of (5) is taken and gives

$$\frac{dY_a}{d\Omega} = \frac{-\alpha \cdot \left(Y^{-1/\alpha} - 1\right)}{\left[1 + \left(Y^{-1/\alpha} - 1\right) \cdot \Omega\right]^{(\alpha+1)}} \tag{6}$$

Equation 6 is the exact expression for the change in apparent yield as a function of the change in defect coverage, however, the expression still contains two unknown terms, the true yield and the absolute defect coverage. To alleviate this problem, an

approximation is made. To simplify the equation, we perform a Taylor series expansion of (6) about the point $\Omega=1$ and consider only the zero order term. Furthermore, the approximation of very high defect coverage allows the substitution of the minimum observed yield (Y_{min}) for the true yield. This approximation is justified by the fact that the known defect level for this device is much less than a fraction of a percent. The approximation gives

$$\left.\frac{dY_a}{d\Omega}\right|_{\Omega=1} = -\alpha \cdot Y_{min} \cdot \left(1 - Y_{min}^{\ 1/\alpha}\right) \qquad (7)$$

where Y_{min} in this case is the observed yield after the applying the full test pattern set of T1 to T5. This approximation is specifically valid when

$$\alpha \cdot \left(1 - Y_{min}^{\ 1/\alpha}\right) \cdot \left(1 - \Omega\right) \ll 1 \qquad (8)$$

Finally, we need to relate a change in fault coverage to a change in defect coverage. In general, the change in defect coverage will be a function of the change in fault coverage and the fraction of defects that behave like the modeled fault. In the case of the experiment reported herein, the stuck-at fault coverage remained constant and the only coverage that changed was the probabilistic bridging fault coverage. Therefore, the change in defect coverage can be modeled as

$$d\Omega = w_{BCE} \cdot d\left(BCE\right) \qquad (9)$$

where w_{BCE} is the fraction of defects that behave as a bridging fault. Substituting (9) into (7) gives

$$dY_a = -\alpha \cdot Y_{min} \cdot \left(1 - Y_{min}^{\ 1/\alpha}\right) \cdot w_{BCE} \cdot d\left(BCE\right) \qquad (10)$$

Equation 10 now relates the observed change in apparent yield to the known change in bridging coverage through a constant term. This equation is only valid when the stuck-at fault coverage is very high and is unchanged in subsequent pattern application as in the case of this experiment. Figure 9 shows the observed change in apparent yield as a function of the change in BCE for each subsequent application of patterns T_2 to T_5. The dashed line is the least squares regression of the data and shows the linear relationship between the change in BCE and the change in apparent yield. From this data it is possible to determine w_{BCE} by

$$w_{BCE} = \frac{m}{-\alpha \cdot Y_{min} \cdot \left(1 - Y_{min}^{\ 1/\alpha}\right)} \qquad (11)$$

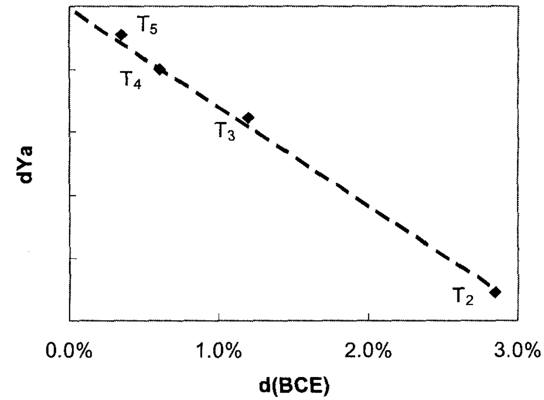


Figure 9 Observed change in apparent yield for each subsequent section of the multiple detect pattern set versus the change in *BCE*. The dashed line is the linear regression of the data.

where m is the slope of the regression. Based on the observed Y_{min} and $\alpha=4$, w_{BCE} is determined to be 8.36%. This is the estimated fraction of the defects that behave as a node-to-node bridging defect. Although α in this case was explicitly determined using a windowing technique, it should be noted that the result does not change significantly over a broad range of clustering. For example, a range of $1 \le \alpha \le \infty$, results in $8.64\% > w_{BCE} > 8.27\%$. Furthermore, using (8), the defect coverage is determined to be 0.418% better with the multiple detect pattern set versus the single detect pattern alone.

The true measure of the impact of multiple detect patterns is the change in defect level. To obtain an expression for this, we take the derivative of (1) with respect to the apparent yield

$$\frac{dD_L}{dY_a} = \frac{Y}{Y_a^2} \qquad (12)$$

The same approximation made earlier is again made such that $Y=Y_a=Y_{min}$, and (10) is substituted in (12)

$$dD_L = -\alpha \cdot \left(1 - Y_{min}^{\ 1/\alpha}\right) \cdot w_{BCE} \cdot d\left(BCE\right) \qquad (13)$$

Equation 13 expresses the change in defect level as the result of a change in the bridging coverage. Again, due to the approximations made, the use of this equation is limited to situations where the inequality of (8) is met.

Table 5 shows the predicted change in defect level based on (13) for two different example conditions. The first condition is an example of a small die size running in a fairly mature process and represents a

A (cm^2)	D0 (cm^{-2})	Y	ΔD_L (DPM)	
			$\Delta\Omega$=-0.42% (without SPP)	$\Delta\Omega$=-0.25% (with SPP)
0.3	0.3	91.5%	368	221
1.4	0.7	41.6%	3290	1976

Table 5 Example DPM increase based on the observed change in defect coverage for the multiple detect pattern set.

reasonable best case scenario. The second condition is an example worst case with a large die running in a fairly early process. The change in defect level is listed for two different changes in defect coverage. The first is the measured change in defect coverage for test sets T_2 to T_5 as reported above. The second case is the equivalent change in defect coverage when devices are screened with SPP. The table provides an example range of the expected DPM increase if multiple detect patterns are not used. Notice that the impact of not employing multiple detect patterns can be significantly mitigated with proper IDDQ outlier screening with SPP.

8. Conclusions

In this paper a new ATPG methodology has been introduced to create patterns that improve the quality of scan tests by maximizing the probability of detecting bridging defects. Two metrics have been introduced to measure the effectiveness of a test set in detecting bridging defects. The patterns expose larger percentage of node-to-node bridges by targeting each fault multiple times in several different ways. Patterns were generated using this methodology and applied to test a 0.18μm LSI Logic ASIC. Volume data was collected and shows that there is very good agreement between the proposed BCE metric utilized by the ATPG tool and actual silicon fallout. The results further show that the measured change in defect coverage is about 0.42% and can be reduced to 0.25% with effective I_{DDQ} outlier screening using Statistical Post-Processing™.

9. Acknowledgements

The authors would like to acknowledge many fruitful discussions with Greg Aldrich, Mark Kassab, Nilanjan Mukherjee, Ron Press and Chen Wang of Mentor Graphics. The author's further acknowledge Mike Chandler of LSI Logic for the failure analysis work.

10. References

[1] V. Krishnaswamy, A. B. Ma, and P. Vishakantiaiah, "A Study of Bridging Defect Probabilities on a Pentium™ 4 CPU", *Proc. ITC*, pp. 688-695, 2001.

[2] S. Chakravarty, A. Jain, N. Radhakrishnan, E. W. Savage, and S. T. Zachariah, "Experimental Evaluation of Scan Tests for Bridges", *Proc. ITC*, pp. 688-695, 2002.

[3] M. R. Grimaila, Sooryong Lee; J. Dworak, K. M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, Jaehong Park, L.-C. Wang, M. R. Mercer, "REDO-random excitation and deterministic observation-first commercial experiment", *Proc. VTS*, pp. 268 – 274, 1999.

[4] J. Dworak, J. Wicker, S. Lee, M.R. Grimaila, K.M. Butler, B. Stewart, L.C.Wang and M.R. Mercer, "Defect-oriented testing and defective part level prediction for commercial sub-micron ICs", *IEEE Design and Test of Computers,* pp.31-41, Jan.-Feb. 2001.

[5] S. Lee, B. Cobb, J. Dworak, M. R. Grimaila and M. R. Mercer "A New ATPG Algorithm to Limit Test Set Size and Achieve Multiple Detections of all Faults", *Proc. DATE*, pp.94-99, 2002.

[6] E. J. McCluskey, Chao-Wen Tseng, "Stuck-fault tests vs. actual defects", *Proc. ITC*, pp.336-342, 2000.

[7] A. Pancholy, J. Rajski, L. J. McNaughton, "Empirical failure analysis and validation of fault models in CMOS VLSI circuits", *IEEE Design & Test of Computers*, Vol. 9, Issue 1, pp.72-83, Mar 1992

[8] R. Madge, M. Rehani, K. Cota and R. Daasch, "Statistical Post-Processing at Wafersort – An alternative to Burn-in and a manufacturable solution to test limit setting for sub-micron technologies", *Proc. VLSI Test Symposium*, 2002

[9] J.T. de Sousa and Vishwani D. Agrawal, "Reducing the Complexity of Defect Level Modeling using the Clustering Effect", *Proc. DATE*, pp. 640-644, Paris, March 2000.

[10] S. C. Seth and V. D. Agrawal. "Characterizing the LSI Yield Equation from Wafer Test Data", *IEEE Trans. on CAD*, CAD-3(2):123-126, April 1984.

[11] V. D. Agrawal, S. C. Seth, and P. Agrawal. "Fault Coverage Requirement in Production Testing of LSI Circuits", *IEEE Journal of Solid State Circuits*, SC-17(1):57-61, Feb. 1982.

[12] T. W. Williams and N. C. Brown, "Defect level as a Function of Fault Coverage", *IEEE Trans. On Computers*, vol. C-30, no. 12, 1981, pp. 987-988.