A New ATPG Algorithm to Limit Test Set Size and Achieve Multiple Detections of all Faults

Sooryong Lee, Brad Cobb, Jennifer Dworak, Michael R. Grimaila and M. Ray Mercer
Department of Electrical Engineering
Computer Engineering Group
Texas A&M University
College Station, Texas 77843-3259
E-mail: mercer@ee.tamu.edu

Abstract

Deterministic observation and random excitation of fault sites during the ATPG process dramatically reduces the overall defective part level. However, multiple observations of each fault site lead to increased test set size and require more tester memory. In this paper, we propose a new ATPG algorithm to find a near-minimal test pattern set that detects faults multiple times and achieves excellent defective part level. This greedy approach uses 3-value fault simulation to estimate the potential value of each vector candidate at each stage of ATPG. The result shows generation of a close to minimal vector set is possible only using dynamic compaction techniques in most cases. Finally, a systematic method to trade-off between defective part level and test size is also presented.

1. Introduction

The objective of testing is filtering defective chips from manufactured ones to reduce the fraction of defective parts that are erroneously sold to customers as defect-free parts. Historically the single stuck-at-fault(SAF) model has been widely accepted as a standard target model to generate a set of test patterns to detect all the SAFs in the circuit. However, it does not account for the necessity of detecting other defect types seen in real manufacturing environments. In fact we cannot achieve 100% defect coverage with test patterns of 100% SAF coverage. On the other hand, the exact mechanisms of various defects are unknown in many cases, and defects are so diverse that accurate universal defect modeling is a very difficult task. With millions of transistors on a single device, it is clear that it is not practical to generate deterministic test sets for all possible defect models.

There is a strong correlation between the number of times a fault site is “observed” and the ability of the corresponding test set to screen out arbitrary defects that occur at that site. Hence, whenever a fault site is observed, probabilistic excitation of non-target defects at that site leads to fortuitous defect detection. Since a conventional automatic test pattern generator($ATPG$) stops at 100% SAF coverage, not all fault sites are observed enough times to ensure sufficient detection of non-target defects. We propose a new $ATPG$ algorithm which guarantees multiple deterministic observation and random excitation of all the fault sites in the circuit.

However, multiple observation of each fault site usually requires a bigger test set size. For real circuits, tester memory space is usually limited, and a significant increase in test pattern set size will exceed tester memory limitations. In such a situation, the time required to apply a test set may increase unacceptably because multiple loading sequences are needed for complete testing. The proposed algorithm tries to focus on generating a minimal test set through a greedy strategy. In fact, a small fraction of all faults dominate the test set size because usually only one of the hard faults is detected by a test pattern. Our approach detects the hard faults and directs additional $ATPG$ effort toward finding tests that detect multiple hard faults. After a minimal test set is generated, we suggest a method to estimate the relationship between test set size and defective part level.

2. Defective part level model

The MPG-D(Mercer, Park, Grimaila and Dworak) model [1] was developed to predict the final defective part level(DPL) using information on the number of times sites are observed and the probability of exciting undetected defects when we make those observations.

This model reflects the fact that SAFs are closely related to real defects, but they only cover part of the defect universe. The model also uses the fact that random excitation and deterministic observation of a site may enable fortuitous detection of defects which are related to that site. Therefore, we can achieve a sufficient confidence level of defect testing by observing each node of the circuit enough times to excite potential defects at that site. Therefore defect-oriented $ATPG$ must generate test patterns which can detect all the sites multiple times and excite randomly during each detection.

$MPG\text{-}D$ assumes that, initially, defects are uniformly distributed across all circuit sites and assigns to each circuit site i its initial contribution to the overall defect level based upon the formula (1). Here Y represents the manufacturing yield.

$$DL_i(0) = \frac{1-Y}{\#\_of\_sites} \tag{1}$$

An iterative approach is used so that the probability of excitation of an as yet undetected defect, given that site i is observed, is an exponentially decreasing function as (3) and the DPL after a new vector is applied is given as (4). For simplicity, every defect is assumed to be associated with only one site in the circuit of interest. After v vectors have been applied, the overall defect level becomes,

$$TotalDL(v) = \sum_{i=1}^{\#\_of\_sites} DL_i(v) \tag{2}$$

$$P_{excite_i} = e^{-\frac{\#obs_i}{\tau}} \tag{3}$$

$$DL_i(v) = \begin{cases} (1 - A * P_{excite_i})DL_i(v-1) \\ \qquad \text{if site } i \text{ was observed by vector } v \\ DL_i(v-1) \\ \qquad \text{otherwise} \end{cases} \tag{4}$$

A and τ are constants which affect the contribution of the current vector to overall DPL by random excitation.

The model has been validated using surrogate bridging fault simulation on both benchmark and commercial circuits. Also, the model matches well with statistical data from commercial chip testing [2]. The quality of our new defect-oriented $ATPG$ will be evaluated based upon this model.

3. Determining the lower bound on test set size

A theoretical lower bound on test set size has been studied. A set of faults is said to be $independent$ if no two faults can be detected by the same vector [3]. This concept leads directly to a lower bound on the size of the required test set. However, the maximal number of independent faults is sometimes an unachievable lower bound for some circuits. Below, we introduce a more realistic measure for evaluating the compactness of a test set.

A fault A is said to be $compatible$ with a fault B if there is a test which can detect both fault A and fault B. Compatibility of two faults can be represented as the intersection of all possible tests for each fault as shown in Figure 1.

We can approximate the compatibility of two faults as the number of successful test generation attempts from an $ATPG$ process targeting both faults for the same test pattern. For some given fault of interest, the summation of compatibilities(SC_i) across all other faults is a reasonable measure of how hard fault i is to detect fortuitously when a test is generated by targeting a different fault. For n faults, SC_i is given as

$$SC_i = \sum_{j=1}^{n} \alpha_j \qquad \begin{array}{l} \alpha_j = 1 \text{ if test exist for fault } i \text{ and } j (i \neq j) \\ \alpha_j = 0 \text{ otherwise} \end{array} \tag{5}$$

Compatibility determines how much compaction of test vectors can be accomplished since the number of faults with minimal compatibility sets a lower bound on the number of tests for a given circuit. Faults with minimal compatibility are closely related to hard-to-detect faults because these faults are not usually fortuitously detected in a test generation session for other target faults.

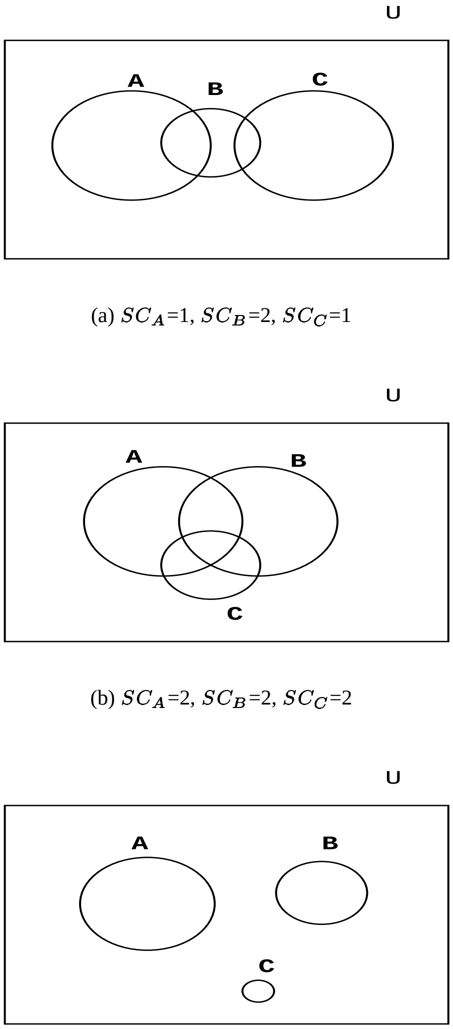
The relationship between faults with minimal compatibility and hard-to-detect faults is analyzed later. To achieve a near-minimal test set, we should focus on faults with minimal compatibility during the entire $ATPG$ process.

4. Defect-oriented ATPG

Conventional $ATPG$ stops generating additional test patterns if 100% SAF coverage is achieved. As noted earlier, some defects may remain undetected even after applying a perfect 100% SAF test set.

We systematically extended the conventional $ATPG$ process in order to further detect defects as shown in Figure 2 and Figure 4.

This new algorithm utilizes 3-value(0,1,X) fault-simulation through successive stages of the $ATPG$ process that produces a single test pattern. With 3-value simulation and partially specified inputs, we predict the number of deterministically detected faults in the case where all requirements for their detection have been satisfied based upon current assignments to primary inputs. During this process, some of the primary inputs remain unspecified, and, therefore, the detection status of some or many of the faults inside circuit will be unknown. Thus, at this stage, there are faults that will or will not be detected depending upon the



(a) SC_A=1, SC_B=2, SC_C=1

(b) SC_A=2, SC_B=2, SC_C=2

(c) SC_A=0, SC_B=0, SC_C=0

Figure 1. Diagram illustrating SC

values that will eventually be assigned to currently unspecified primary inputs. We say these faults are potentially detected. The number of potentially detected faults is a good secondary estimator of the contribution of partially specified vectors to overall deterministic detection.

But this estimator is not exact since conflicts of testing conditions between different potentially detectable faults can exist before final binary values are specified for all primary inputs. To minimize the uncertainty introduced by these potential conflicts, our algorithm checks for the locally incompatible potential faults in advance. We consider three kinds of conflict types. For example, for a NAND gate,

1. Both the stuck-at 0 and the stuck-at 1 fault at the same site cannot be simultaneously detected.

2. Only one gate input stuck-at 1 fault can be detected for any given pattern.

3. Both the stuck-at 1 input fault and stuck-at 1 output fault cannot be simultaneously detected.

Note that an equivalence-collapsed fault set is assumed here. Rules for other gate types can be derived in a similar way. Even though these conflict rules are utilized only locally at individual gates, they significantly enhance the accuracy for our detection estimator.

Various combinations of weighting factors for the deterministic detection component and the potential detection component can be used for the optimum result. A greed_metric(GM) is given as follows for n faults.

$$GM = \sum_{i=1}^{n} (\omega e^{-\frac{\alpha_i * M}{N}} + e^{-\frac{\beta_i * M}{N}}) \qquad (6)$$

α_i= number of detection of fault i

β_i= number of potential detection of fault i

ω= weighting deterministic detection

$\frac{M}{N}$= weighting least detected fault

Note that the number of detections is used to prioritize the least detected faults. In our implementation N is given as the target number of observations and M is set at 2.

The algorithm consists of two parts, random pattern generation and deterministic pattern generation.

Let us assume the circuit has n inputs and the computing system has an m-bit word size. Note that the employed fault simulation is a Parallel Pattern Fault simulator(PPF) combined with Critical Path Tracing(CPT) [4]. In the conventional scheme, a parallel pattern fault simulator is not fully utilized since most $ATPG$ algorithms create a single test pattern for a target fault. However, our scheme

```
repeat
   begin
      assign all $PI$s $X$ value
      repeat
         begin
            randomly select $k$ $PI$s with $X$ value
            assign $2^k$ vector combinations to selected $PI$s
            3-value parallel fault-simulation
            choose $PI$ value with maximum $GM$
         end
      until $PI$s fully specified or 0 potential detection
   end
until escape-condition is met
```

Figure 2. Random part of greedy algorithm

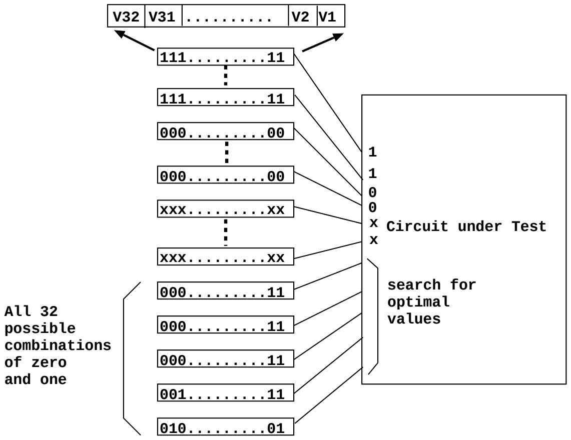


Figure 3. A snapshot of PIs

utilizes the extensive parallel processing capability of the PPF. Moreover CPT simulates faults implicitly, which means the cost of simulation does not depend on fault dropping. This feature is beneficial for keeping the fault dictionary information. In the random part, first k(=$\log_2 m$) primary inputs(PI) of n are selected randomly. m combinations of parallel patterns are applied to selected PIs and the remaining PIs have 'X' values. Figure 3 shows a snapshot of PIs in the circuit(for the case where $k = 5$). The 3-value parallel pattern fault simulation result shows m different GMs for each vector. The vector with maximum GM is selected as the final vector for k PIs. This process is repeated until all not-yet-determined PIs are fully specified. Therefore $\lceil \frac{n}{\log_2 m} \rceil$ iterations are required to compute our near-optimal test vector.

The random approach described above is applied before switching to a deterministic method presented below. The decision to change from the random approach to the deterministic approach will be controlled by an escape-condition. In our implementation, the ineffectiveness of the random approach is measured in the following way. If $EC > \alpha(= 0.9)$, the algorithm escapes the random loop.

```
repeat
   begin
      assign all $PI$s $X$ value
      search least detected fault
      generate test for chosen fault
      repeat
         begin
            randomly select $k$ $PI$s with $X$ value
            assign $2^k$ vector combinations to selected $PI$s
            3-value parallel fault-simulation
            choose $PI$ value with maximum $GM$
         end
      until $PI$s fully specified or 0 potential detection
   end
until all non-redundant faults are detected $N$ times
```

Figure 4. Deterministic part of greedy algorithm

$$EC = 1 - \frac{GM_i}{GM_{i-1}} \tag{7}$$

‘

GM_{i-1}= GM value of previous vector

GM_i= GM value of current vector

If the improvement of a new vector is minimal, then the deterministic method that targets specific faults to generate patterns is used. The deterministic method is similar to conventional ATPG in this sense, but the least detected fault is selected as the target fault to get more defect coverage.

After a partial pattern is generated for the target fault, the remaining unspecified('X') PIs will be explored using the same method as the random method. The test generation process for each target fault stops if no more potential detection is possible. This not only saves computing time but also introduces another compaction possibility. The final set of test patterns may have unspecified 'X' values. Various static compaction techniques might also be applied for more minimization.

One important difference between conventional $ATPG$ and the new one is the searching criteria. In conventional $ATPG$, the efficiency of searching is a priority to reduce $ATPG$ time so we always choose what are estimated to be the easiest excitation and propagation paths for a target node. Normally these paths are unique. Therefore if the same fault is targeted again, we get exactly the same vector which will not detect any additional defects. Furthermore, this efficiency-oriented approach does not always produce the best defect-coverage vector since the search is biased for $ATPG$ efficiency. Our new $ATPG$ always tries to randomize the searching process for both excitation and propagation. Not only can we get different patterns for each run, but we also produce statistically diverse patterns for the same fault site which may detect additional defects fortuitously.

5. Experimental results

5.1. Test set size

We implemented this algorithm in the C language on a SUN ULTRA 5 workstation. Results for several ISCAS85 benchmark circuits are shown in Table 1. Data in the second column is the number of test patterns that detect all non-redundant faults at least 15 times. Because no other published works have reported test set sizes for multiple detections, we compare our results with published results reporting test pattern length for single detection of every fault in the circuit. Thus, we must scale down the length of our test set by the number of detections achieved. The average test set size in the third column is obtained by dividing data in the second column by 15. These results are compared with previous results published in [5]. The data in the column [5] is one of the best results from compaction algorithms of conventional ATPG which employed extensive dynamic and static compaction technique. The data in the column L.B. is the maximal number of independent faults that determine a theoretical lower bound on the test set size.

In an ideal world, test sets that guarantee 15 detections of each fault would have a size considerably less than 15 times corresponding test set sizes that achieve just one detection per fault. In contrast, this data shows that, for most circuits, the test set size grows about linearly with the number of detections. We believe that this is true because the number of tests to detect a few hard faults determines the total test length. One counter-example is c6288 where our algorithm achieves sub-linear test pattern length growth for increasing numbers of fault detections. In c2670 our algorithm performs worse (test length growth is greater than linear). Perhaps, static compaction techniques would improve results for this circuit. On the other hand, this super-linear growth may be artifact of the circuit's character. Perhaps, because our test vectors excite each site randomly for every observation, slightly bigger average test set size is to be expected.

5.2. The impact of hard faults

To evaluate the influence of hard-to-detect faults on test set size, we did a more detailed analysis for the c432 circuit. First, we identified 43 faults as "hard" because they are only detected 15 times by the completed test set. Among 507 test patterns, only 75 patterns are not involved in the detection of any of the 43 hard faults. (Eighty-five percent of all test patterns are required for this relatively small subset of all faults in the circuit.) Each of these 43 hard-to-detect faults was targeted 100 times in order to estimate their approximate SC values. Figure 5 shows that minimal compatibility exist between these hard-to-detect faults. In other words, the overlapping area of possible tests for each of the 43 faults is minimal. None of these hard faults is, on average, detected simultaneouly with two other hard faults. In order to achieve even minimal compaction of these hard faults, we must be willing to expend significant test pattern generation effort and computer time.

circuit	test size	avg. size	[5]	L.B.	CPU time(sec.)
c432	505	33.7	29	27	292.1
c499	793	52.9	52	52	153.2
c880	338	22.5	21	12	229.6
c1355	1274	84.9	84	84	5674.6
c1908	1648	109.9	106	99	1563.9
c2670	962	64.1	45	42	9357.6
c6288	144	9.6	14	6	1813.8

Table 1. Experimental results for ISCAS85 benchmark circuits

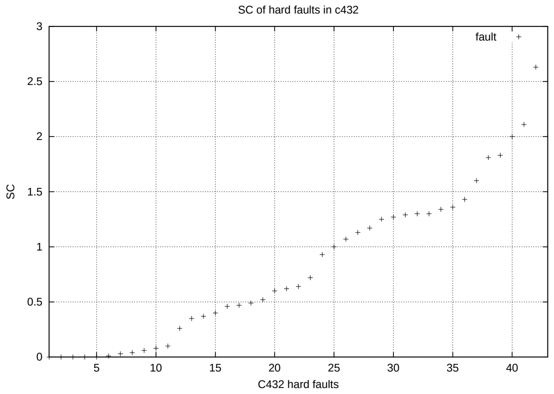


Figure 5. SC of c432 hard faults

5.3. Trade-off between defective part level and test set size

Even though we got a relatively compact test pattern set with minimal defective part level, the final test set size is still large. Hence, we propose a systematic method to trade-off between test set size (or tester memory) and test quality. The resulting test patterns are evaluated in terms of their contribution to the overall DPL reduction using the MPG-D model, and they are sorted in descending order. Figure 6

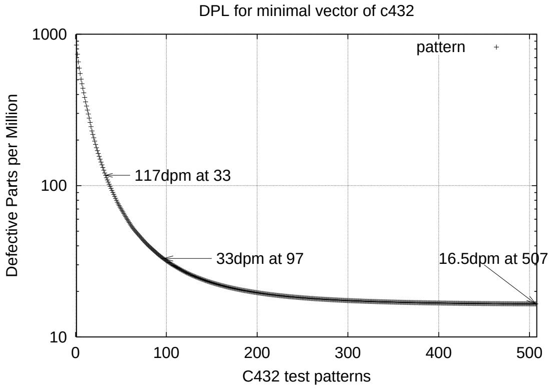


Figure 6. DPL of minimal vector for c432

shows the final DPL graph based on this approach for the c432 circuit. If the objective DPL is specified, we can truncate the full test set to produce a corresponding shorter test set. Similarly, if the objective test set size is specified, we can predict a final DPL for that test set. In summary, we can intelligently trade off between test quality and test cost using this graph.

Note that a factor of two increase in defective part level (from 16.5 to 33 dpm) allows a reduction in test set size from 507 to 97(more than a factor of five). To relate this result to industrial circuits, we note that about 33 test patterns are required to obtain 100% stuck-at fault coverage. Therefore, for a commercial circuit, it is reasonable to expect that about three times the number of patterns required to attain 100% fault coverage will achieve an excellent defective part level. This test set may fit in the available tester memory. In contrast, a test pattern length of 507 patterns for c432 corresponds to an increase by a factor of 15 relative to current test pattern lengths for commercial circuits, and the chance that such a large test pattern set would all fit in the available tester memory is very small.

6. Conclusion

A new algorithm that observes and randomly excites all the fault sites in a circuit multiple times is proposed. To get minimal test sets, a greedy approach is adopted and a 3-value fault simulation is extensively used. The result shows this dynamic compaction technique can achieve a close-to-minimal test vector set size that also reduces overall defective part level dramatically. Finally, a systematic method to trade-off between test quality and test set size is presented. This method can be used as a guideline for a practical testing in a commercial environment.

References

[1] 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* Jan.-Feb. 2001, pp.31 41

[2] M.R. Grimaila, S. Lee, J. Dworak, K.M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, L.C. Wang and M.R. Mercer, “REDO-Random Excitation and Deterministic Observation-First Commercial Experiment”, *Proc. 1999 VLSI Test Symposium*, 1999, pp. 268-274

[3] S.B. Akers, C. Joseph and B. Krishnamurthy, “On the role of independent fault sets in the generation of minimal test sets”, *Proc. 1987 Int. Test. Conf.*, 1987, pp. 1100-1107

[4] H.K. Lee and D.S. Ha, “An efficient, forward fault simulation algorithm based on the parallel pattern single fault propagation”, *Proc. 1991 Int. Test. Conf.*, 1991, pp. 946-955

[5] S. Kajihara, I. Pomeranz, K. Kinoshita and S.M. Reddy, “Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits”, *IEEE Tran. on CAD* vol. 14 No. 12, Dec. 1995, pp. 1496-1503