REDO — Random Excitation and Deterministic Observation — First Commercial Experiment *

Michael R. Grimaila[1], Sooryong Lee[1], Jennifer Dworak[1], Kenneth M. Butler[2],
Bret Stewart[2], Hari Balachandran[2], Bryan Houchins[2], Vineet Mathur[2],
Jaehong Park[1], Li-C. Wang[1], and M. Ray Mercer[1]

[1]Department of Electrical Engineering, Computer Engineering Group
Texas A&M University, College Station, Texas 77843-3128
E-mail: mercer@ee.tamu.edu

[2]Semiconductor Group, Texas Instruments
8505 Forest Lane, MS 8645, Dallas, Texas 75243
E-mail: kenb@ti.com

Abstract

For many years, non-target detection experiments have been simulated by using AND/OR bridges or gross delay faults as surrogates. For example, the defective part level can be estimated based upon surrogate detection when test patterns target stuck-at faults in the circuit. For the first time, test pattern generation techniques that attempt to maximize non-target defect detection have been used to test a real, 100% scanned, commercial chip consisting of 75K logic gates. In this experiment, the defective part level for REDO- based patterns was 1,288 parts per million lower than that achieved by DC stuck-at based patterns generated using today's state of the art tools and techniques.

1 Introduction

We use the term *defects* to denote actual flaws in an integrated circuit, which introduce erroneous operation for some input sequence. Similarly, the term *fault* denotes the abstract defect models used as targets to produce test patterns. Finally, *surrogates* are defect models (which are not targeted during test pattern generation) but are used to quantify the non-target defect detection of a given test pattern set.

Although stuck-at fault detection is widely accepted in industry as a key test quality figure of merit, it does not account for the necessity of detecting other defect types seen in real manufacturing environments [BUTL90][MA95]. Other researchers have addressed this problem by using various "enhanced fault models" during the ATPG process [FERG91]. In this case, the fault simulation engine is modified to allow the simulation of the "enhanced fault model" to more accurately emulate real defects encountered in the manufacturing process. Unfortunately, using defect models during the ATPG process is too costly in both time and memory. Multiple fault models with multiple testing methods have also been proposed and studied [MAX92a][MAX92b]. Still other researchers have noted that targeting stuck-at faults may detect many bridges and vice versa [MEI74][MILL88]. We extend these notions by noting that any logic defect may be fortuitously detected by tests targeted for stuck-at faults.

In fact, extremely valuable information is discarded during the fault simulation phase of the traditional ATPG process. Specifically, each site's fault detection profile is lost in modern fault simulators because they use fault dropping for time/space efficiency. However, 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 defects which occur at that site. In an attempt to quantify the defect reduction, we constructed a new defective part level model, **MPG**, that uses each individual site's stuck-at fault detection profile to predict how well a test vector set can detect manufacturing defects. The MPG model is named for its inventors M. R. **M**ercer, J. **P**ark, and M. R. **G**rimaila.

Our research differs from existing work in the way it

*This work was supported by the Texas Advanced Technology Program – Project No. 036327-152.

attempts to reduce the overall defective part level. In our method, we use a standard stuck-at fault model-based ATPG, but we modify the fault simulator to keep track of each node's stuck-at fault detection profile. The sum of the stuck-at one detects and the stuck-at zero detects for a given site is called the site "observation count." By dynamically targeting test vector generation toward the least observed sites and using random decision ATPG, we effectively reduce the overall defective part level of the device. When a fault is detected multiple times given different excitation or observation conditions, we have a higher probability of detecting most of the various defects involving that fault site. A key benefit of this approach is that no defect models except for stuck-at faults are required to direct the ATPG process [WMW96].

We applied a traditional test pattern set consistent with current best commercial practice and our enhanced test pattern set to an actual integrated circuit with 75,000 logic gates. An analysis of the results of the experiment supports our conjecture that increasing each site's observability significantly reduces the overall defective part level for the device.

In the case of this particular chip, the traditional test pattern generation time was less than ten minutes. In contrast, the chip design was months. Thus increased test pattern generation effort represents an appropriate engineering choice. Many additional CPU cycles may be used, but these are very inexpensive and run in parallel with other design activities. This additional effort is not at all likely to increase the total design cycle. Our optimized test pattern selection process required several days, but this was a very small fraction of the total design time.

2 The Role of Excitation and Observation in Defect Detection

Generating tests using the stuck-at fault model requires deterministic excitation and deterministic observation. In Figure 1, a simple circuit is used to illustrate the steps required to generate a test for A stuck-at 1. In this case, A is set to a 0 to excite the fault and S is set to a 1 to allow the fault, D, to propagate through NAND gate N1. Since the signal S is set to a one, the output of inverter I1 is a 0. This zero insures that the output of gate N2 is a one, which allows the D on the output of N1 to propagate to the output of N3 as a D bar. In this case, it does not matter what value is assigned to node B and the resulting value at the output of I2 because this value is blocked from propagating through N3 due to the 0 on the output of I1. Given the conditions that A is equal to 0 and that S is equal to 1, we find that the probability of both the excitation and the observation of fault A stuck-at 1 is equal to 1.

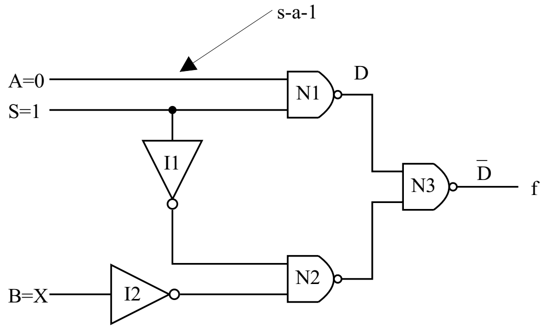


Figure 1: Example circuit with a test for node A s-a-1

If we apply the same test vector to the circuit shown in Figure 2 which contains an OR bridge between node A and node B, we find that the value of node B will determine if the OR bridge is detected or not. If B is equal to 0, the OR bridge is not detected. If B is equal to a 1, then the OR bridge is detected. Given the conditions that A is equal to 0 and that S is equal to 1, we find that the probability of excitation of the OR bridge defect in this example is $\frac{1}{2}$. Thus, we see that the excitation for a bridge fault when using a stuck-at fault model for test generation can be probabilistic.

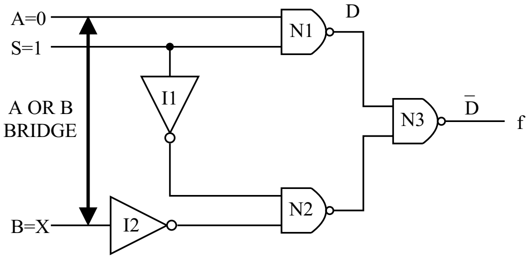


Figure 2: Example circuit with A OR B bridging fault

If we apply the same test vector to the circuit shown in Figure 3 which contains an OR bridge between node A and node BN, we find that the value of node BN will determine if the OR bridge is detected or not. If BN is equal to 0, the OR bridge is not detected. If BN is equal to a 1, then the OR bridge is detected. Given the conditions that A is equal to 0 and that S is equal to 1, we find that the probability of excitation of the OR bridge defect in this example is $\frac{1}{2}$. Also note that if two different chips have the defects of Figure 2 and Figure 3, then both tests must be applied in order to reject both of the defective chips.

Figure 4 shows a much more generalized comparison of test generation for any arbitrary point in the

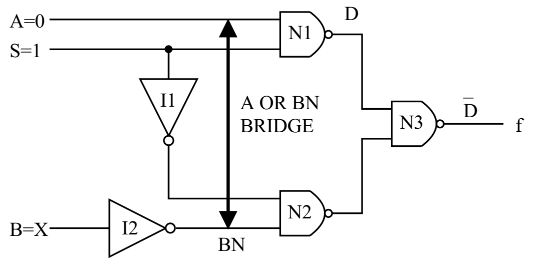
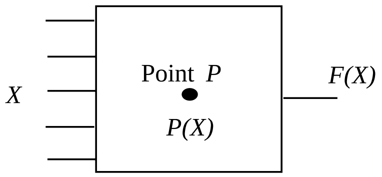


Figure 3: Example circuit with A OR BN bridging fault

network. For the traditional method, let χ_E be a member of the subset of all possible tests such that $p(\chi_E) = 0$ and the fault P s-a-1 is excited (Capitals indicate faulty values and the lower case indicates good values.) If $\partial f/\partial P(\chi_E) = 1$, then P s-a-1 will be observed and detected. The REDO test approach selects a set of inputs, χ_O such that $\partial f/\partial P(\chi_O) = 1$, and some of these tests probabilistically excite (and detect) defects at P. Thus, for the case of a stuck-at fault, both the excitation, $P(\chi)$, and observation conditions, $\partial f/\partial P(\chi)$, are deterministic. In contrast, the test for a defect at node P will have the same deterministic observation, but the probability of defect excitation depends upon the particular character of that defect.



$$T_{P/1}(\chi_E) = \overline{P}(\chi_E) \times \frac{\partial f}{\partial P}(\chi_E)$$
$$T_{DEFECT}(\chi_O) = \left\{ \begin{array}{c} \text{Probabilistic} \\ \text{Defect Excitation} \end{array} \right\} \times \frac{\partial f}{\partial P}(\chi_O)$$

Figure 4: A comparison between deterministic testing for a stuck-at fault and probabilistic testing for a defect

3 The MPG Defective Part Level Model

We now introduce a defective part level model, called MPG, which uses the number of times each fault site is observed to predict the defective part level. For this work, we assume a constant probability of excitation given that the site is observed, P_{EXCITE}, and a constant probability of the existence of a defect, P_{DEFECT}.

We now examine each component of the MPG model shown in Figure 5. The number of times site i is observed, $\#OBSi$, is the sum of stuck-at one and stuck-at zero fault detections for that site. Notice that the probability of not detecting the defect, $(1 - P_{EXCITE})^{\#OBSi}$, is reduced as a site is observed more frequently. The expression $(1 - P_{EXCITE})^{\#OBSi} \times P_{DEFECT}$ calculates the probability of escape, that is, the probability that a defect occurs but is not detected. This model assumes statistical independence between: (1) the probability of excitation for each individual observation, and (2) the probability of detection and the probability of occurrence for defect at the given site. If we subtract the probability of escape from one for each site, we obtain the probability that an escape does not occur for that site. The product over all sites of this probability is the probability that no escape occurs for the complete circuit. The defective part level is then determined as the ensemble probability that at least one escape occurs.

$$DL \cong 1 - \prod_{i=1}^{Sites} [1 - (1 - P_{EXCITE})^{\#OBSi} \times P_{DEFECT}]$$

$P_{EXCITE} \equiv$ Probability a defect is excited given that its site is observed (Assumed constant)
$(1 - P_{EXCITE})^{\#OBSi} \equiv$ Probability defect at site i is never detected
$P_{DEFECT} \equiv$ Probability of a defect at site i
P_{DEFECT} is calculated from:
$Yield = (1 - P_{DEFECT})^{\#Sites}$

Figure 5: The **MPG** defective part level model

It is very simple to consider several special cases and observe that the MPG model's predictions are logical in those instances. For example, consider that no testing at all is done so that there are no observations. In this case, the defective part level reduces to exactly $(1 - Yield)$. (This result follows exactly based upon the last line in Figure 5.) Similarly, as the number of observations at all sites approaches infinity, the predicted defective part level approaches zero.

This model has several advantages when compared with existing stuck-at fault-based defect level models. For example, in the traditional models, the defective part level is predicted to be zero when the stuck-at fault coverage reaches 100% [SETH84][WILL81]. In contrast, this new model exponentially approaches but never actually reaches a defective part level of zero. The MPG model uses observation information which is calculated as part of stuck-at fault simulation, and its predictions vary based upon the number of observations which occur at any given site. For the defects remaining undetected as the test pattern application process proceeds, chances are that they are hard to excite. Thus, using a constant (but relatively small) probability of excitation for all defects results in a conservative upper bound for the actual defective part level achieved. Alternatively, the probability of excitation can be modeled using a more sophisticated function which involves the number of observations at each individual site and refined estimates of defect occurrence.

4 An Initial Test Generation Method for Defect Level Minimization

Based upon the MPG defective part model given above, a family of new test generation methods have been developed and evaluated. In each case, tests are produced by traditional ATPG algorithms targeting traditional stuck-at faults. The only non-standard requirements on the ATPG algorithm are: (1) that the test pattern generation decision process be random so that if the same fault is used twice as a target, the resulting two tests are extremely unlikely to be identical (i.e. are random samples from the set of all possible test patterns for the fault), and (2) that the number of detections of each fault over the entire test pattern set be computed during fault simulation. In fact, if two identical patterns are produced, one is removed in a post-processing step after ATPG and prior to the optimization described in Section 7.

In contrast to traditional fault selection methods (such as testing for each undetected fault at least once), we select faults to be targeted in such a way as to maximize the number of times that sites are observed. More specifically, since those sites which are least often observed make the dominant contribution to the defective part level, our objective is to maximize the number of observations of ”hard to observe” sites. This means that many stuck-at faults will never be used as targets — because they are often detected by tests targeted at other faults. In contrast, some stuck-at faults will be targeted many times because they are located at sites which are only rarely observed.

Figure 6 shows a simplified flow chart for the initial test pattern generation approach used. The key differences from standard procedures are: (1) that the least detected fault is always selected for processing next, and (2) fault detection statistics are saved which record how many times each fault has been detected to date. Note that this is orders of magnitude less information than what is collected for a fault dictionary. Initially, all faults have been detected exactly zero times. From this set, one is randomly selected and used as a target to produce the first test pattern. After fault simulation, many faults may have one detection; of the faults with zero detections, another is randomly selected as the next target. Eventually, every fault has been detected at least once, but the process continues by finding the set of faults which have been least detected and randomly selecting one of them as the next target. In some cases, no test is produced because an upper bound on ATPG time is exceeded, and the corresponding fault is temporarily removed as a candidate target to avoid excessive ATPG times.

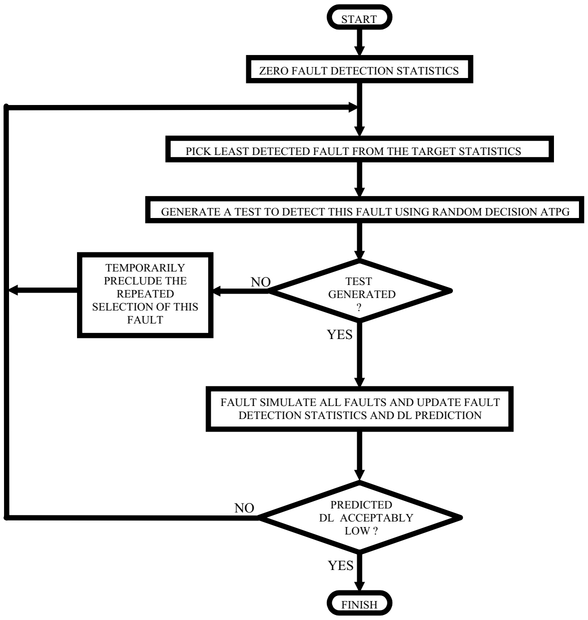

START
ZERO FAULT DETECTION STATISTICS
PICK LEAST DETECTED FAULT FROM THE TARGET STATISTICS
GENERATE A TEST TO DETECT THIS FAULT USING RANDOM DECISION ATPG
TEMPORARILY PRECLUDE THE REPEATED SELECTION OF THIS FAULT
NO
TEST GENERATED ?
YES
FAULT SIMULATE ALL FAULTS AND UPDATE FAULT DETECTION STATISTICS AND DL PREDICTION
NO
PREDICTED DL ACCEPTABLY LOW ?
YES
FINISH

Figure 6: Dynamic least detected fault targeting algorithm

As usual, there are many variations on the basic theme. In the section that follows, three such variations are described, and their performance is compared in a defect (surrogate) simulation experiment.

5 Predictions for C432 Based upon Surrogate Simulation

In order to study the effectiveness of test pattern sets produced based upon the MPG defective part level model, a simple set of experiments were conducted using the ISCAS benchmark circuit — C432. All stuck-at one and stuck-at zero faults, after equivalent fault collapse, were used as the basic fault set, and statistics on the number of detections of each fault were monitored during the test pattern generation and fault simulation process [LEE93]. The yield was assumed to be 96.7% (to match the actual data from the commercial part). A total of 45,000 AND and

OR bridges between lines in the C432 circuit were modeled, and the defective part level which was determined based upon the number of bridges which remained undetected is shown on the Y axis. Three methods were used to produce test pattern sets, and the results are shown for comparison in Figure 7.

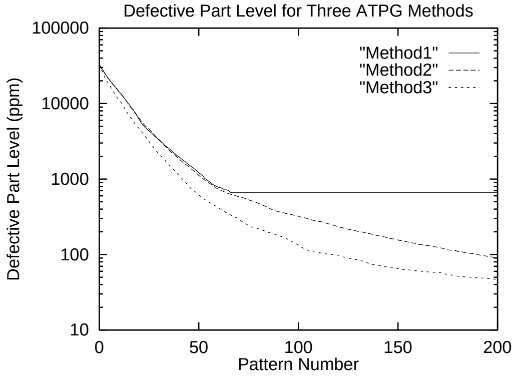


Figure 7: Defective part level versus pattern number for three different ATPG methods

Method 1 corresponds to traditional industrial practice today. A test pattern set is produced by targeting one currently undetected fault, doing fault simulation, and dropping all detected faults from future consideration as targets. This process is repeated until every detectable fault has been either explicitly targeted and detected or fortuitously detected by a test for some other target fault. At the end of 70 test patterns, the fault coverage is 100%, and the process terminates. The resulting defective part level is 661 parts per million. Method 2 corresponds to the dynamic least detected fault targeting described in the flow chart of Figure 6. In this case, the test generation process was terminated after 200 test patterns were generated. This number of patterns was used because it was about three times the number of patterns required to attain 100% fault coverage. (Tester limitations for the commercial part, described in the next section, restricted the number of patterns in that case to about three times the number required to achieve nearly 100% fault coverage, and we wanted the two test pattern lengths to be “relatively equal.”) Observation of Figure 7 shows that the resulting defective part level for this approach is 90.1 parts per million.

Method 3 is similar to Method 2 except that after the ATPG algorithm has produced a test pattern, the test may contain one or many unspecified primary input values (assigned as ”X” values). We produce up to 32,768 different patterns by random assignments of 0 and 1 to the X’s and fault simulate all of them. The test pattern that maximizes the number of fault detections is selected as the test pattern to be included in the final test pattern set. In all, 200 test patterns were generated. This approach was used to understand the level of improvement which could be expected if the test pattern generation effort was increased by several orders of magnitude. It attempts to find an “approximate bound” for the best possible test pattern set targeted exclusively based upon fault detection statistics. The resulting defective part level is 46.8 parts per million.

6 Predicted Defective Part Levels for the Commercial Chip

The part used in this study was a commercial chip consisting of more than 75,000 two input NAND equivalent logic gates. Best practice stuck-at fault testing methods were employed in the standard test flow. Two test pattern sets were applied to the chip of interest. One was a standard test pattern set produced by a commercially available ATPG tool with a stuck-at fault coverage in excess of 97% and a test pattern length of 3,000 scan chain loads (designated in Figure 8 as “Commercial”). The second test pattern set was the optimized set described in the next section consisting of 3,000 scan chain loads (designated in Figure 8 as “Research”). All of the state elements could be independently controlled and observed via the scan chain.

Figure 8 shows two example predictions produced by the MPG defective part level model using only the results from scan-based stuck-at fault testing on the digital portion of the device. Here, the X-axis corresponds to fault number (from 1 to 80,000) where the faults are ordered from lowest defective part level contributor (largest number of observations) to highest defective part level contributor (smallest number of observations). The Y-axis shows the resulting predicted cumulative defective part level for the set of all faults from the smallest contributor to the fault corresponding to that X-axis position. The value for Y at the extreme right side of the graph corresponds to the defective part level for the chip as predicted by the MPG model. Note that: (1) a significant component of the defective part level is contributed by a relatively small number of faults located to the far right side of the X-axis (these are the targets of opportunity for superior test pattern selection methods), and (2) that the MPG defect levels predicted for our new test pattern generation method are significantly below those for current best test pattern generation methods.

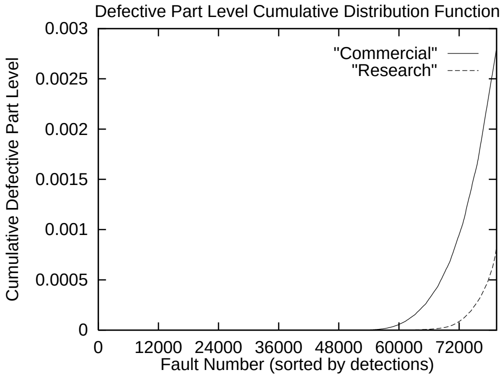


Figure 8: MPG defective part level model predictions versus fault number for the commercial device

7 An "Optimized" Test Generation Method for Defect Level Minimization

We now examine the methods used to select the optimized test pattern set. Figure 9 shows the conceptual procedure used to produce the optimized test pattern set, and Figure 10 shows the procedure as a simplified flow chart.

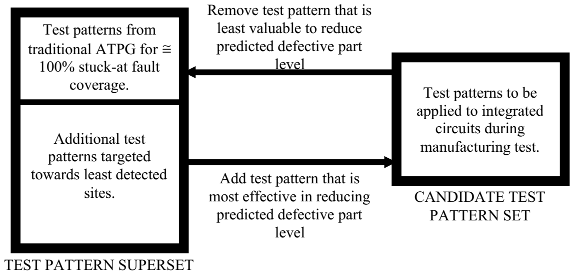

Test patterns from traditional ATPG for ≅ 100% stuck-at fault coverage.
Additional test patterns targeted towards least detected sites.
TEST PATTERN SUPERSET
Remove test pattern that is least valuable to reduce predicted defective part level
Add test pattern that is most effective in reducing predicted defective part level
Test patterns to be applied to integrated circuits during manufacturing test.
CANDIDATE TEST PATTERN SET

Figure 9: Optimize test vector selection algorithm

First, a SUPER TEST PATTERN SET was produced as described in the flow chart of Figure 6. Unfortunately, the length of this set exceeded the 3,000 pattern limit by about a factor of four. Thus, it was necessary, because of tester memory limitations, to select an OPTIMIZED SUBSET corresponding to the set of 3,000 test patterns that produce the lowest defective part level according to the MPG model. For successive steps, the defective part level reduction predicted for every pattern in the CANDIDATE OPTIMIZED SET was evaluated, and the least effective pattern was removed. Then, the defective part level reductions predicted by the MPG model for every test pattern in the SUPER TEST PATTERN SET (which did not already exist in the current CANDIDATE OPTIMIZED SET) were compared, and the most effective new pattern was added to form the new CANDIDATE OPTIMIZED SET. The process of removing the weakest test pattern from the CANDIDATE SET and adding the strongest pattern from the SUPER TEST PATTERN SET was repeated until no significant difference in predicted defective fault level was achieved. We used the resulting test pattern set as our OPTIMIZED test pattern set to be applied during manufacturing test. Note that the information required for this process was similar to that which is produced in a fault dictionary.

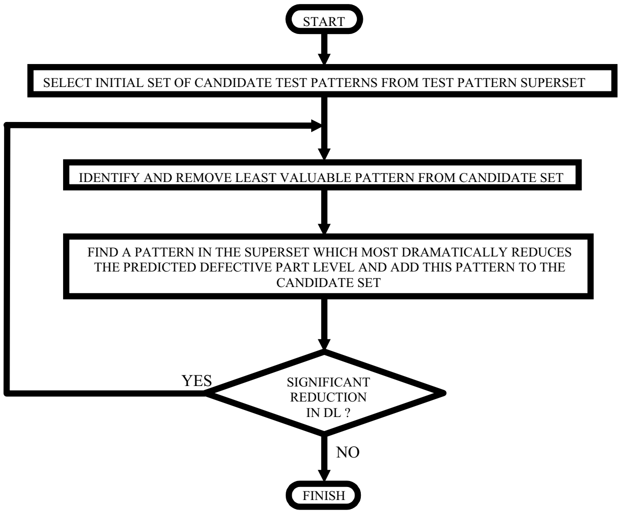

START
SELECT INITIAL SET OF CANDIDATE TEST PATTERNS FROM TEST PATTERN SUPERSET
IDENTIFY AND REMOVE LEAST VALUABLE PATTERN FROM CANDIDATE SET
FIND A PATTERN IN THE SUPERSET WHICH MOST DRAMATICALLY REDUCES THE PREDICTED DEFECTIVE PART LEVEL AND ADD THIS PATTERN TO THE CANDIDATE SET
YES
SIGNIFICANT REDUCTION IN DL ?
NO
FINISH

Figure 10: Minimum defective part level test vector selection algorithm

8 Actual Relative Defective Part Levels for the Commercial Chip

To protect confidential yield information, we will not report on the number of manufactured die which failed parametric tests, but 6,986 die passed all parametric tests and were subjected to both a traditional DC test pattern set and the REDO OPTIMIZED DC test pattern set described above. In the end, 220 defective die of the 6,986 total were detected by current best commercial test practice, and 229 defective die were detected from the same set using the OPTIMIZED test pattern set. With this sample size, each defective die corresponded to 143 defective parts per million (1,000,000 / 6,986). The resulting defective part level (Y-axis) versus applied test pattern number (X-axis) plots are shown in Figure 11. The X axis range was limited to 1500 patterns because no defective parts were identified after pattern 1500 by either test set. Since we have no idea how many defective die were erroneously passed by the OPTIMIZED set, we arbitrarily assume that two such defective die escaped. (If this number is changed, the plots will move up or down on the Y-axis, but the spacing will always terminate with the same difference.) Because this part is a mixed signal device, analog testing after the digital tests may further reduce the final defective part level.

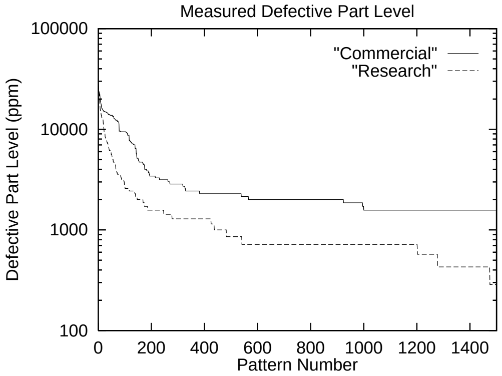


Figure 11: Measured defective part level for commercial and research test vectors

9 Conclusions

We have described the MPG defective part level model and the REDO method, which produced OPTIMIZED test pattern sets for digital logic circuits. Both surrogate simulation of the C432 benchmark circuit and actual testing of a commercial part using best current practice indicate the superiority of the REDO testing method for the reduction of defective part levels in digital integrated circuits. This is the first of several testing experiments, and future experiments are scheduled with much larger die sample sizes. We also are studying techniques to reduce the test pattern generation and optimized test pattern selection times.

The REDO improvement simply involves using traditional ATPG and fault simulation tools in a new way. By maximizing the deterministic observation of defect sites in the network (as determined from traditional stuck-at fault simulation) and relying upon probabilistic defect excitation, significant improvements in test pattern efficiency have been achieved. In particular, considering stuck-at fault testing alone, the REDO tests resulted in 1,288 fewer defective parts per million. Obviously, this does not mean that this chip is being shipped at a DPPM level of 1,288 since a number of other tests are applied which also have the effect of lowering the defective part level.

Acknowledgments: The authors would like to express appreciation for the constructive and stimulating comments and observations by Rohit Kapur and T. W. Williams related to this work. Additionally, we would like to thank D.S. Ha for use of the academic ATPG tool ATLANTA and Don Ross of Mentor Graphics for the use of FASTSCAN, a commercial ATPG tool.

References

[BUTL90] Butler, K.M. and Mercer, M.R., “The influences of fault type and topology on fault model performance and the implications to test and testable design,” *Proc. 27th ACM/IEEE Design Automation Conference 1990*, pp. 673-678.

[FERG91] Ferguson, F.J. and Larrabee, T., “Test pattern generation for realistic bridge faults in CMOS ICs,” *Proc. International Test Conference 1991*, pp. 492-499.

[LEE93] Lee, H.K. and Ha, D.S., “On the generation of test patterns for combinational circuits,” *Technical Report No. 12, Department of Electrical Engineering,* Virginia Polytechnic Institute and State University, 1993.

[MA95] Ma, S., France, P., and McCluskey, E.J., “An Experimental Chip to Evaluate Test Techniques: Experimental Results,” *Proc. International Test Conference 1995*, pp. 663-672.

[MAX92a] Maxwell, P.C. and Aitken, R.C., “IDDQ Testing as a Component of a Test Suite: The Need for Several Fault Coverage Metrics,” *Journal of Electronic Testing (JETTA)*, Vol. 3, 1992, pp. 305-316.

[MAX92b] Maxwell, P.C., Aitken, R.C., V. Johansen, and I. Chiang, “The Effectiveness of IDDQ, Functional, and Scan Tests: How Many Fault Coverages Do We Need?” *Proc. International Test Conference 1992,* pp. 168-177.

[MEI74] Mei, K.C., “Bridging and stuck-at faults,” *IEEE Trans. on Computers*, Vol. C-23, no. 7, 1974.

[MILL88] Millman, S.D. and McCluskey, E.J., “Detecting bridging faults with stuck-at test sets,” *Proc. International Test Conference 1988*, pp. 773-783.

[SETH84] Seth, S.C. and Agrawal, V.D., “Characterizing the LSI Yield from Wafer Test Data,” *IEEE Trans. on CAD*, Vol. CAD-3, 1984, pp. 123-126.

[WILL81] Williams, T.W. and Brown, N.C. “Defect level as a function of fault coverage,” *IEEE Trans. on Computers*, Vol. C-30, no. 12, 1981, pp. 987-988.

[WMW96] Wang, L-C., Mercer, M.R., and Williams, T.W., “Using Target Faults to Detect Non-Target Defects,” *Proc. International Test Conference 1996*, pp. 629-638.