THE EFFECTIVENESS OF DIFFERENT TEST SETS FOR PLAs

Peter C. Maxwell

Hewlett-Packard Company,
Design Technology Laboratory,
3172 Porter Drive,
Palo Alto, CA 94304, USA.

Hans-Joachim Wunderlich

University of Karlsruhe,
Institute of Computer Design & Fault Tolerance,
Postfach 6980, D-7500, Karlsruhe 1,
Federal Republic of Germany.

ABSTRACT

It has been theoretically demonstrated that the single stuck-at fault model for a PLA does not cover as many faults as the single crosspoint model. What has not been demonstrated is the real relative effectiveness of test sets generated using these models. This paper presents the results of a study involving presenting a number of test sets to fabricated PLAs to determine their effectiveness. The test sets included weighted random patterns, of particular interest owing to PLAs being random resistant. Details are given of a method to generate weights, taking into account a PLA's structure.

1. INTRODUCTION

PLAs are used extensively as modules within a chip and are incorporated into both standard cell and custom designs. Much has been written regarding testing PLAs and the fault models employed to generate tests. In order to increase coverage to include multiple faults, many approaches require design for testability of the PLAs, involving varying amounts of hardware overhead compared to an unmodified PLA. A good survey of these may be found in Agarwal (1985) and more recent examples are Reddy & Ha (1987), Upadhyaya & Saluja (1988) and Liu & McCluskey (1988). Other approaches deal with an unmodified PLA, which will not have any overheads associated with it, although will not have as high a guaranteed fault coverage. Notwithstanding this, in many applications where speed and/or area are critical, the overheads involved in producing an easily testable design are considered unacceptable, with the result that efforts have been directed at investigating the generation of effective test sets. It is this class of PLAs towards which this paper is directed.

The traditional fault model used in test pattern generation schemes is the stuck-at model, and early PLA test generation systems used this model (e.g. Muehldorf & Williams, 1977; Eichelberger & Lindbloom, 1980). In this approach, the PLA is effectively modelled as a collection of gates, and tests are generated by assuming single stuck faults occur on the nodes of this gate equivalent circuit.

A potential problem with the stuck-at model is that it does not explicitly take into account *bridging faults*. The structure of a PLA, with its long runs in metal, is such that this type of failure is likely to happen. There is no a priori guarantee that a stuck-at test set will detect bridging faults. Further, it is not feasible to analytically determine the effectiveness of a stuck-at test at detecting these faults. Consequently, it is unpredictable how well a stuck-at test set will detect bridging faults.

More recent approaches (e.g. Wei & Sangiovanni-Vincentelli, 1985; Robinson & Rajski, 1988) use the crosspoint fault model (Ostapko & Hong, 1979), which is concerned with the potential presence or absence of programming transistors in the arrays. Tests are then generated to detect these conditions. As with the stuck-at model, bridging failures are not explicitly taken into consideration. However, it is possible to analyze the efficiency of crosspoint tests with regard to many bridging failures (Ostapko & Hong, 1979) with the result that single crosspoint tests are guaranteed to detect many bridging failures also. Further, stuck-at faults can be shown to be a subset of crosspoint faults.

In order to overcome some of the theoretical deficiencies of the stuck-at model, Cornelia & Agrawal (1989) have proposed a "conditional stuck-at" model, which requires that a generated vector detect not only a targetted fault, but also produce a specific binary value on another specified line in the gate level circuit description. By doing this the fault coverage is extended to guarantee inclusion of single crosspoint faults and single bridging faults.

Adopting a more general approach, bridging faults and multiple stuck-at faults can be shown to be equivalent to multiple crosspoint faults (Ligthart & Stans, 1989) so that a multiple crosspoint test set will achieve the highest coverage of those faults believed to be the most commonly occurring. However, the single crosspoint model is the one adopted for test generation systems, and there have been investigations regarding the multiple fault coverage of the test sets generated (Agarwal, 1980; Rajski & Tyszer, 1986). Coverage of multiple faults is good, but not guaranteed.

2024/90/0000/0628$01.00©1990 IEEE

Compared to a single stuck-at test set, a crosspoint test set size is larger because of the larger number of faults considered. The gate level model building effectively ignores any "crosspoints" which have no transistors since these will not contribute to any gate. However, larger test sets may have a better chance of detecting other faults, in particular stuck open or transition faults (Waicukauski, 1988) so this is not necessarily a disadvantage.

One of the problems of investigations regarding multiple fault coverage is the implicit assumption about the structure of the PLA. In particular, no attention is paid to the fact that the PLA may be folded. In this case many crosspoints are no longer physically present, so that a subset of the faults considered are no longer realistic since they cannot occur. Further to this, due to the resultant geometries, many other faults, notably bridging, will not realistically occur (without causing catastrophic failure for which no sophisticated test is required). Consequently, the equivalent multiple crosspoint fault will also not occur.

The result is that for fabricated PLAs, the situation is complex and not readily amenable to general analysis. Although investigations such as inductive fault analysis (Ferguson & Shen, 1988) can suggest likely failure modes it is not clear how reliably simulation results can predict the effectiveness of a test set in detecting defective chips.

Because of the tradeoffs involved between test set size, potential fault coverage and (to a lesser extent) generation time, a useful comparison is to determine which (if either) method generates tests which are more effective in screening real chips. Although bridging failures have been mentioned above, there are many other types of failures possible (Ferguson & Shen, 1988), and it is not feasible to attempt to take them all into account, either in fault simulation or in theoretical analyses. What has been noticeably absent in all discussions regarding test sets and fault models is experimental data derived from testing real, fabricated PLAs.

Given a variety of test sets generated using different fault models, it is desirable to know the relative effectiveness of each set in screening defective chips. This paper reports the results of such a comparison, which can be realistically carried out only by testing fabricated silicon, since any fault simulator is limited to the fault models supported.

2. TEST SETS CONSIDERED FOR EVALUATION

1. Stuck-at tests generated using the *ATG* program. This is an internal program which is designed for general logic, and consists of a random pattern phase followed by a D-algorithm phase. The program is not specifically oriented towards PLAs, with the result that test generation times are longer than could be achieved with a more specific approach. A gate level description of a PLA is required.

2. Test sets derived using the *platest* program. This is another internal program which is heuristically based and uses a combination of stuck-at and crosspoint models.

3. Two crosspoint test sets, one generated using the Berkeley *platypus* program (Wei & Sangiovanni-Vincentelli, 1985) and the other using the McGill University *planet* program (Robinson & Rajski, 1988).

4. A weighted random pattern test set generated by the *implant* program (a modification of Wunderlich, 1985). These are of particular interest since PLAs are known to be resistant to conventional random patterns.

3. GENERATION OF WEIGHTED RANDOM PATTERNS

Usually PLA's are not considered to be random pattern testable, since product terms which depend on a large number of literals are activated with a low probability. In many cases random pattern testability cannot be improved by using weighted random patterns corresponding to a single set of weights, because different product terms might have contradictory weight requirements. For multilevel combinational circuits this problem can be solved by computing multiple sets of weights (Waicukauski, 1988, Wunderlich, 1988). This approach can be applied to PLA's as well.

Computing multiple sets of weights is based on computing fault detection probabilities. Let $X := (x_1, ..., x_m) \in [0,1]^m$ denote a set of weights corresponding to the input probabilities, i.e. input $i \in I$ is set to a logical "1" with probability x_i. For each fault $f \in F$ the detection probability $p_f(X)$ is the probability that f is detected by one randomly selected input pattern corresponding to X.

It is shown in Wunderlich (1988) that for a given number N of random patterns a set of weights X is optimal, if the expression

$$J_N^F(X) := \sum_{f \in F} e^{-N p_f(X)}$$

is minimized. Optimizing procedures are also presented there. Moreover, it is shown that multiple sets of weights can be computed by dividing F into subsets F_1, F_2 such that

$$\| \operatorname{grad} (J_N^{F_1}(X)) \| + \| \operatorname{grad} J_N^{F_2}(X)) \|$$

is maximal. Two sets of weights X^1 and X^2 are computed by minimizing $J_N^{F_1}(X^1)$ and $J_N^{F_2}(X^2)$, respectively. Details can be found in Wunderlich (1988). Obviously, a fast method for computing fault detection probabilities $p_f(X)$ and conditional probabilities $p_f(X,0_i)$, $p_f(X,1_i)$ is required. The latter is needed to compute the first partial derivative

$$\frac{dp_f(X)}{dx_i} = x_i \,(p_f(X,1_i) - p_f(X,0_i))$$

and to compute the gradients.

Two major changes are required in order to adopt this approach for PLA's:

- a more complex faulty behavior must be considered, e.g. bridging faults or crosspoint faults,
- the two-level description of a PLA should be used for estimating fault detection probabilities more efficiently.

3.1 Fault Injection

Let $I := (i_1, ..., i_m)$ and $O := (o_1, ..., o_n)$ be the primary inputs and outputs, respectively. The function of a PLA is described by a cover

$$\boldsymbol{C} := (C^1, ..., C^k) \text{ of cubes } C^j := (c^j_1, ..., c^j_m, c^j_{m+1}, ..., c^j_{m+n}).$$

Each cube corresponds to a product term P_j (Brayton, 1984). For $1 \leq h \leq m$ we have c^j_h = "-", if input variable i_h does not appear in product term P_j, $c^j_h = 0$, if i_h is negative in P_j, and $c^j_h = 1$, if i_h is positive in P_j. For $m+1 \leq h \leq m+n$ we have c^j_h = "-", if product term P_j does not contribute to output o_h, and $c^j_h = 1$, if P_j does. Hence each cover describes a multi-output boolean function.

A fault f is injected into the PLA by modifying its cover to $\boldsymbol{C}_f$. For each stuck-at fault, crosspoint fault and bıidging fault the modification can be done with linear effort (Daehn, 1986).

3.2 Estimating Fault Detection Probabilities

A fault f is detected, if a pattern sets an output variable of $\boldsymbol{C}_f$ to "TRUE" and the same variable to "FALSE" in $\boldsymbol{C}$, or vice versa. Computing the probability of this event is a #-complete problem and in general requires an infeasible effort. In Wunderlich (1987) a polynomial sampling procedure was presented, which still requires a large computing time. We now present a more efficient approach.

Let $\boldsymbol{C}$ be a cover and let $\boldsymbol{C}^j := \{C \in \boldsymbol{C} \mid c_j = 1\}$ be the subset of all cubes which contribute to output o_j. For a cube $C \in \boldsymbol{C}$ define

$$t_C(X) := \prod_{i=1}^{m} t^i_C,$$

where $t^i_C = 1$ if c_i = "-", $t^i_C = x_i$ if $c_i = 1$, and $t^i_C = 1 - x_i$ if $c_i = 0$.

For each fault f define the detection function

$$d_f : [0,1]^m \rightarrow [0,1],\ d_f(X) = 1 - \prod_{j=m+1}^{m+n} (1 - d^j_f(X)$$

where

$$d^j_f(X) = [1 - \prod_{C \in \boldsymbol{C}^j} (1 - t_C(X))] \prod_{C \in \boldsymbol{C}^j_f} (1 - t_C(X)) + [1 - \prod_{C \in \boldsymbol{C}^j_f} (1 - t_C(X))] \prod_{C \in \boldsymbol{C}^j} (1 - t_C(X))$$

Now let $Y \in \{0,1\}^{\mathrm{m}}$ be a tuple of constant probabilities $y_i = 0$ or $y_i = 1$. Then Y corresponds to a certain test pattern Y', and we have $d^j_f(Y) = 1$ if and only if the fault f is detectable at output j by the pattern Y'. Hence we have

$d_f(Y) = 1 <==> Y'$ is a test pattern for f.

For nonconstant values, $d_f(\mathrm{X}) \approx p_f(\mathrm{X})$ can be considered as an estimation of the detection probability of f. The estimation error induced by $d_f(X)$ decreases the more the values of X are biased to 0 or 1.

The formula for $d_f(X)$ is linear in the size of the PLA's. In general we have

$$\left| \bigcup_{f \in F} \boldsymbol{C}_f \cup \boldsymbol{C} \right| << \sum_{f \in F} | \boldsymbol{C}_f | + | \boldsymbol{C} |,$$

and the efficiency in evaluating $d_f(X)$ is increased drastically by storing the values of $t_C(X)$ for all cubes C involved and by storing the values of $\prod_{C \in \boldsymbol{C}^j} (1 - t_C(X))$.

Based on the estimations $p_f(X) \approx d_f(X)$, the well-known methods for computing multiple sets of weights are applied. The results presented in this paper are obtained by random patterns corresponding to up to 20 sets of weights. The underlying fault model was the crosspoint fault.

4. COMPARISON OF TEST SETS

4.1 PLAs and Test Set Sizes

The test vehicles used were 5 PLAs which form part of a larger chip. Each were individually scanned and therefore able to be tested independently. Table 1 summarizes the PLAs and the number of test vectors applied in each test set. The PLAs were fabricated using a double metal 1µ CMOS process.

4.2 Generation Times

Table 2 shows test generation times for *ATG, platest, platypus* and *implant*. Times for the first three are in CPU seconds on an HP 9000/350 while the last is on a Sun 3/50. Times for *planet* are not available at the time of writing.

5. RESULTS AND CONCLUSIONS

At the time of writing, results are available from testing 2256 chips. For the PLAs ADCU, ADCL, ESDI and STMD, no difference was observed in the performance of any of the test sets. Any failure detected by one set was also detected by all other sets.

For the case of TTG, some variation occurred, which is summarised in Table 3. This table shows the number of failed vectors for each test set for each of 7 die for which *variation* occurred, together with the total number of rejected die by each test set.

Probably due to the exclusiveness of each test set (there were very few vectors in common), the detected faulty chips of other sets were not subsets of those detected by *ATG*. In particular, the highlighted column shows one chip which passed all sets except the weighted random patterns, where it failed on one vector. This is most likely to represent a delay fault, although this has not been confirmed.

Although these variations occurred, they are not statistically significant at the 95% confidence level. Thus it may be concluded:

PLA NAME	ADCL	ADCU	STMD	ESDI	TTG
No. of inputs	14	14	21	50	44
No. of outpus	52	64	27	39	54
No. of terms	65	53	134	98	110
No. of FETs	2740	3552	2903	3173	4232
atg (HP)	246	309	403	138	561
platest (HP)	296	379	762	542	1522
planet (McGill)	346	422	611	250	716
platypus (Berkeley)	297	381	654	287	766
implant (Karlsruhe)	5000	3100	6951	4901	4151

Table 1. PLAs and test set sizes.

	ADCL	ADCU	STMD	ESDI	TTG
ATG	660	1260	3840	840	3099
platest	12	17	95	181	264
platytpus	6	8	29	18	46
implant	834	1874	6653	760	1652

Table 2. Test generation times in CPU seconds.

	#1	#2	#3	#4	#5	#6	#7	total rejects
ATG	1	0	3	30	12	66	60	321
platest	0	0	0	0	0	1	0	316
planet	6	0	0	0	1	9	15	320
platypus	0	0	0	0	1	118	6	318
implant	2	1	0	0	0	0	23	318

Table 3. Number of vectors causing rejection in the 7 die which showed *variation* between test sets.

1. The single stuck-at test set performs as well as either crosspoint test set. Given the completeness of the conditional stuck-at model, this result is not completely surprising, since many of the necessary conditions are likely to be met.

2. Despite the fact that PLAs are random pattern resistant, a set of multiple weights for weighted random patterns can be computed in reasonable time, and the resultant test set performs as well as any of the more conventional test sets.

6. ACKNOWLEDGEMENTS

This work would not have possible without the considerable time and effort spent by Hai Vo-Ba and Jim Cooke at HP's Colorado Integrated Circuits Division in reformatting the test sets suitable for their testers, in application of the tests, and in collecting and summarising the data. Thanks are also due to Janusz Rajski at McGill University for providing the *planet* test set.

7. REFERENCES

1. V.K. Agarwal (1980), "Multiple Fault Detection in PLAs", *IEEE Trans. on Comput.*, **C29**, pp 518-522.

2. V.K. Agarwal (1986), "Easily Testable PLA Design", in *VLSI Testing*, T.W. Williams (ed.), Elsevier, North Holland, pp 65-93.

3. L. Blank (1980), *Statistical Procedures for Engineering, Management and Science*, McGraw-Hill.

4. R. Brayton et al (1984), *Logic Minimization Algorithms for VLSI Systems*, Kluwer, Boston.

5. O.E. Cornelia & V.K. Agarwal (1989), "Conditional Stuck-at Fault Model for PLA Test Generation", VLSI Design Laboratory, McGill University.

6. W. Daehn (1986), "A Unified Treatment of PLA Faults by Boolean Differences", *Proc. 23rd Design Automation Conference*, pp. 334-338.

7. E.B. Eichelberger & E. Lindbloom (1980), "A Heuristic Test-Pattern Generator for Programmable Logic Arrays", *IBM Journal of Res. & Dev.*, **24**, January, pp 15-22.

8. F.J. Ferguson & J.P. Shen (1988), "Extraction and Simulation of Realistic CMOS Faults Using Inductive Fault Analysis", *Proc. IEEE Test Conf.*, pp 475-484.

9. G.K. Garner (1984), "PLA Test Generation", *Proc. Design Technology Conf.*, pp S4-3-1 - S4-3-10.

10. M.M. Ligthart & R.J. Stans (1989), "A Fault Model for PLAs", *Proc. 1st European Test Conf.*, pp 252-260.

11. D.J. Liu & E.J. McCluskey (1988), "Design of Large Embedded CMOS PLAs for Built-in Self Test", *IEEE Trans. on CAD*, **7**, pp 50-59.

12. E.I. Muehldorf & T.W. Williams (1977), "Optimized Stuck Fault Test Pattern Generation for PLA Macros", *Proc. IEEE Test Conf.*, pp 89-101.

13. D.L. Ostapko & S.J. Hong (1979), "Fault Analysis and Test Generation for Programmable Logic Arrays", *IEEE Trans. on Computers*, **C-28**, pp 617-626.

14. J. Rajski & J. Tyszer (1986), "The Influence of Masking Phenomenon on Coverage Capability of Single Fault Test Sets in PLAs", *IEEE Trans. on Comput.", **C-35**, pp 81-85.

15. S.M. Reddy & D.S. Ha (1987), "A New Approach to the Design of Testable PLAs", *IEEE Trans. on Comput.*, **C-36**, pp 201-211.

16. M. Robinson & J. Rajski (1988), "An Algorithmic Branch and Bound Method for PLA Test Pattern Generation", *Proc. IEEE Test Conf.*, pp 784-795.

17. S.J. Upadhyaya & K.K. Saluja (1988), "A New Approach to Design for Built-in Self Testing PLAs for High Fault Coverage", *IEEE Trans. CAD*, **7**, pp 60-67.

18. J.A. Waicukauski & E. Lindbloom (1988), "Fault Detection Effectiveness of Weighted Random Patterns", *Proc. IEEE Test Conf.*, pp 245-255.

19. J. A. Waicukauski et al (1989), "A Method for Generating Weighted Random Test Patterns," *IBM J. Res. & Dev.*, **33**, pp 149-161.

20. R.-S. Wei & A. Sangiovanni-Vincentelli (1985), "PLATYPUS: A PLA Test Pattern Generation Tool", *Proc. 22nd Design Automation Conf.*, pp 197-203.

21. H.-J. Wunderlich (1985), "PROTEST: A Tool for Probabilistic Testability Analysis", *Proc. 22nd Design Automation Conf.*, pp 204-211.

22. H.-J. Wunderlich (1987), "The Random Pattern Testability of Programmable Logic Arrays", *Proc. ICCD*, pp. 682-685.

23. H.-J Wunderlich (1988), "Multiple Distributions for Biased Random Test Patterns", *Proc. IEEE Test Conf.*, pp. 236-244.