DOT: New Deterministic Defect-Oriented ATPG Tool

Jaan Raik, Raimund Ubar, Joachim Sudbrock
Tallinn University of Technology, Estonia
{jaan|raiub}@pld.ttu.ee

Wieslaw Kuzmicz, Witold Pleskacz
Warsaw University of Technology
{wbk, wap}@imio.pw.edu.pl

Abstract

A method is proposed for combinational deterministic test pattern generation using a uniform functional fault model for combinational circuits. This includes an approach, which allows to find the types of faults that may occur in a real circuit and to determine their probabilities. Additionally, a defect-oriented deterministic test generation tool was developed (DOT), and the experimental data obtained by the tool for ISCAS'85 benchmarks are presented. It was shown that 100% stuck-at fault tests covered only about 80-90% physical defects. The main feature of the new tool is its ability to reach 100% defect efficiency for the given set of defects by proving the redundancy of not detected defects. An interesting conclusion of the experiments is also that up to 25 % of the defects cannot be covered by any voltage test approaches.

1. Introduction

As the complexity of digital circuits is dramatically increasing while the demands on test quality and reliability are getting higher for most products, two main trends to face these problem can be observed: Applying hierachical models to use high level modelling thus decreasing the test comlexity, and using defect oriented approaches. A combination of these two trends will therefore be necessary in the future.

Traditional very popular stuck-at fault (SAF) model has not withstood the test of time. It has been shown that high SAF coverage cannot quarantee high quality of testing [1-3]. The reason is that not every defect can be modelled as a SAF fault.

The types of faults that can be observed in a real gate depend not only on the logic function of the gate, but also on its physical design. These facts are well known [2-5] but usually ignored in engineering practice. In earlier works on layout-based test generation techniques [4, 5] the whole circuits were analysed as single blocks. Such an approach is computationally expensive and thus highly impractical as a method of generation of tests for real VLSI designs.

In this work we characterise physical defects in library cells, determine their probabilities and then use this information for defect oriented test generation at the logic level. We verify the functionality of analysed gates or complex gates for all possible defects and find the actual functions performed, using either transistor-level simulation or an analytical approach based on solving Boolean diffential equations. The defect characterisation process may be computationally expensive, but it is performed only once for every library cell. In other words, we replace in the cells the abstract logic fault models like SAF with realistic physical defect models.

The defect lists of the library components embedded in the circuit are extended by additional physical defects in the close surrounding of the component to take into account also defects outside of components. For these defects additional characterization should be carried out by a similar way as for the library cells.

In some cases the assumption can be made that the majority of defects occur inside the cells and not in the routing between them. Such assumption would not be realistic in the case of older CMOS technologies with two levels of metal and very dense routing. However, in state-of-the-art deep submicron technologies still only one or two levels are used inside cells but 6 or more levels of metal are available for routing. More routing levels means lower sensitivity to defects.

Proceedings of the European Test Symposium (ETS'05)
1530-1877/05 $20.00 © 2005 **IEEE**

Routing between the cells is less dense and various nodes are routed at various metal levels. As a result, probability of shorts outside cells is significantly reduced and therefore do not have to be predicted using the same accuracy as for defects within the library cells. Applying this hierachical approach in predicting the defect probabilities also allows semiconductor manufacturer to predict defect probabilistics within library cells with high accuracy using company confidential information about the layout or defect statistics, while e.g. ASIC designers predicting defects within the interconnects will not have access to this information but also do not need it.

In [6] a new approach was introduced for hierarchical defect simulation based on defect preanalysis for components, and using the results of preanalysis in higher level fault simulation. Here, we generalize this approach by introducing a functional fault model as a method for mapping faults from one hierarchical level to another. Based on this approach, a hierarchical algorithm for defect-oriented deterministic test generation is developed and implemented. The functional fault model in a form of a set of logical conditions (constraints) allows to represent the defects in components and in the communication networks by the same technique.

Then we propose a methodology which allows to find the types of logic faults that may occur in a real circuit, to determine their probabilities of occurrence, and to find the input test patterns that detect these faults. We compare the results obtained in this way with the results of testing of the same circuits by the sequences of test patterns based on the conventional SAF fault model. Experiments were carried out for the ISCAS'85 benchmark circuits.

The paper is organized as follows. In Section 2 we present a new method of parametric fault modeling for carrying out defect analysis in digital circuits, and define the functional fault model as interface for mapping defects from transistor level to logic level. Section 3 presents the results of a probabilistic analysis of defects by simulating them on the transistor level. In Section 4 we describe a new defect-oriented deterministic test generation algorithm. Section 5 presents experimental results, and finally, in Section 6 we draw the conclusions of this research.

2. Parametric defect modeling

In this Section we present a new general fault model for describing and modelling arbitrary physical defects first, in the components of a circuit and in the routing between the components.

Consider a Boolean function

$$y = f(x_1, x_2, \ldots, x_n)$$

implemented by an embedded component C in a digital circuit. Introduce a Boolean variable d for representing a given defect in the component or in the surrounding of the component, which may affect the value y by converting the Boolean function f into another function

$$y = f^d(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots x_p).$$

Here, in general case, the variables $x_{n+1}, \ldots x_p$ belong to the neighbourhood layout of the component which will influence the function y in the presence of defect d.

Introduce now for the block C a generic parametric function

$$y^* = f^*(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_p, d) = \bar{d} f \vee d f^d \qquad (1)$$

as a function of a defect variable d, which describes the behaviour of the component simultaneously for the both, fault-free and faulty cases. For the faulty case, the value of d as a parameter is equal to 1, and for the fault-free case $d = 0$. The solutions of the Boolean differential equation

$$W^d = \frac{\partial y^*}{\partial d} = 1 \qquad (2)$$

describe the conditions which activate the defect d on a line y. The parametric modelling of a given defect d by equations (1) and (2) allows us to use the constraints (the solutions of $W^d = 1$) in defect-oriented test generation. To find W^d for a given defect d we have to create the corresponding logic expression for the faulty function f^d either by logical reasoning or by carrying out directly defect simulation, or by carrying out real experiments to learn the physical behaviour of different defects.

For example, assume there is a short inside the transistor circuit in Fig. 1 described by the function

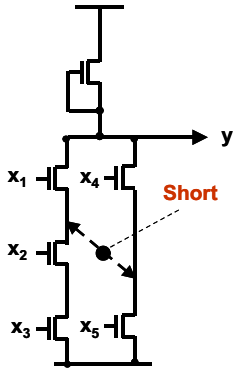

Figure 1. Transistor circuit with a short

Proceedings of the European Test Symposium (ETS'05)
1530-1877/05 $20.00 © 2005 IEEE

$$y = x_1 x_2 x_3 \vee x_4 x_5 .$$

The short changes the function of the circuit as follows:

$$y^d = (x_1 \vee x_4)(x_2 x_3 \vee x_5) .$$

Using the defect variable d for the short, we create a generic Boolean differential equation and solve it as follows:

$$\frac{\partial y^*}{\partial d} = \frac{\partial((x_1 x_2 x_3 \vee x_4 x_5)\bar{d} \vee (x_1 \vee x_4)(x_2 x_3 \vee x_5)d)}{\partial d} =$$

$$= x_1 \overline{x_2 x_4 x_5} \vee \overline{x_1 x_3 x_4 x_5} \vee \overline{x_1 x_2 x_3 x_4} \, \overline{x_5} = 1$$

From the equation three possible solutions follow: T = {10x01, 1x001, 01110}. Each of them can be used as a test pattern for the given short. On this contra-example, it is easy to show the inadequacy of the stuck-at fault (SAF) model for testing the transistor level faults. For example, the set of five test patterns 1110x, 0xx11, 01101, 10110, 11010 which test all the stuck-at faults in the circuit does not include any of the possible test solutions from the set T for detecting the short.

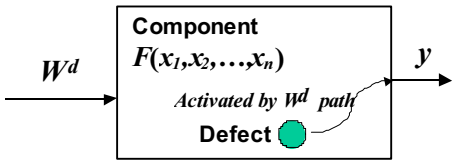

Figure 2. Functional fault model for a physical defect

The described method represents a general approach to map an arbitrary physical defect onto a higher (in this case, logic) level. A physical defect in a component can be represented by a logical constraint $W^d = 1$ to be fulfilled for activating the defect to the output of the component (Fig.2). The event of erroneous value on the output y of a functional component can be regarded as a SAF taking place at the logical constraint $W^d = 1$.

Table 1. Activating conditions for different defects

No	Defect	Conditions W^r
1	SAF $x_k \equiv 0$	$x_k = 1$
2	SAF $x_k \equiv 1$	$x_k = 0$
3	Short between x_k and x_l	$x_k = 1$, $x_l = 0$
4	Exchange of lines x_k and x_l	$x_k = 1$, $x_l = 0$, or $x_k = 0$, $x_l = 1$
5	Delay fault on the line x_k	$x_k = 1$, $x'_k = 0$, or $x_k = 0$, $x'_k = 1$

Some examples of the conditions W^d for different type of defects (where SAF is a particular extreme case) are given in Table 1 (here x_k is the observable variable, and x'_k is the variable at the previous time moment).

3. Probabilistic defect modeling

In the following we consider one kind of physical defects in CMOS gates - shorts between conducting regions. This is one of the most important sources of faults in CMOS digital circuits. However, the methodology can be extended to other types of physical defects as well (e.g. breaks, opens etc.). The defect characterisation was performed for a 0.8μm technology.

A short is a piece of extra conducting material that connects a pair of separate conducting regions in the integrated circuit. This affects the connectivity of the circuit: two separate electrical nodes become connected. It is intuitively obvious that probabilities of shorts depend on the layout of the circuit. Conducting regions that are adjacent to one another are more susceptible to shorts than regions that are separated by a large distance. We assume that every defect that results in a short can be approximated by a circle. To estimate the probabilities of shorts between pairs of nodes we use the concept of critical area for shorts [3]. The critical region for shorts is such a region in the circuit that, if the center of a defect of a given radius R is located anywhere inside the critical area, a short between two adjacent conducting regions occurs (see Fig. 3).

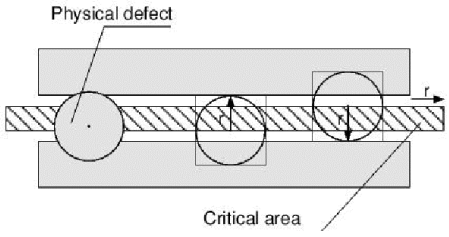

Figure 3. concept of critical area

Its size depends on the shapes and locations of the conducting regions that will be shorted and is a function of the defect radius R. The radii of defects are randomly distributed and can be characterised by a probability density function $Pdf(R)$. This function was modelled in the following way [10]:

Proceedings of the European Test Symposium (ETS'05)
1530-1877/05 $20.00 © 2005 IEEE

$$P_{df} = D_{oi} * \begin{cases} \dfrac{2(p_i - 1) * R}{(p_i + 1) X_{oi}^2} \forall 0 < R < X_{oi} \\ \dfrac{2(p_i - 1) X_{oi}^{p_i - 1}}{(p_i + 1) * r^{p_i}} \forall X_{oi} < R \end{cases}$$

Where i is the type of conducting layer, D_{oi} is the density of physical defects and pi and X_{oi} are modelling parameters. P_i is set to 3, X_{oi} to 20% of the minimal distance between shapes of a given conducting layer and D_o to 10 defects/cm^2

We assume that the probability of a short between a pair of conducting regions that correspond to a pair of electrical nodes is proportional to the critical area for these two regions. The critical area for shorts is a function $P_S(R)$ of the defect radius R. Since the defect radii exhibit a random distribution, the product $Pdf(R) * P_S(R)$ integrated over the range of R where $P_S(R) > 0$ can be taken as the measure of the total probability of shorts between a given pair of nodes. Applying a Poisson based yield model the probability of a short is given by the following formula:

$$P_t = 1 - \prod_{i=1}^{N} \exp\left[- \int_0^{\infty} \mathrm{P_s}(\mathrm{R})\, \mathrm{P_{df}}(R) dR \right]$$

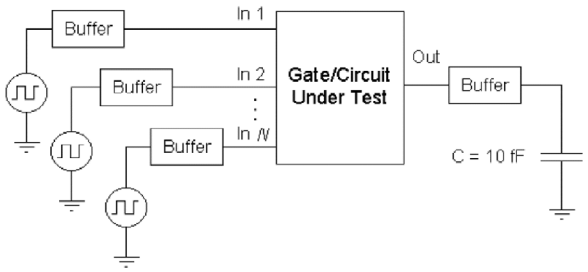

Figure 4. Testbench circuit schematic diagram

The first step in identification of logic faults and their probabilities is to calculate P_t for all pairs of conducting regions representing electrical nodes. If for a given pair P_t is comparatively small, this pair of nodes is very unlikely to be shorted and is therefore not taken into account. For the pairs that can be shorted the logic faults are determined. To carry out the characterisation of the defects, a testbench was implemented (Fig. 4) and the circuit was simulated electrically using SPICE. The channel widths of both NMOS and PMOS transistors were 10µm, the channel lengths were 0.8µm. This way shorted NMOS transistors were stronger than shorted PMOS, creating 'wired AND' shorts, when at least one of the shorted nodes is an input. The shorts were represented by resistors. We tried several resistance values in the range from 0.01 Ω to 200 Ω. The gate behaviour at the logic level did not depend on this resistance. The waveforms obtained from the simulation allow to determine the actual logic function performed by the faulty circuit. Every possible combinational input was applied to the library cell inputs. No sequential test patterns were applied, since all the combinationally redundant defects were proved to be redundant also in the sequential sense. In this way the functional faults that result from shorts were identified and their probabilities were determined. This procedure and the software developed for this purpose are described in more detail elsewhere [7]. Methodology which allows to obtain the defect tables was described in [8].

4. Deterministic test generation algorithm

In order to automatically generate test patterns for defects in CMOS technology, a new deterministic algorithm (Fig. 5) was developed and implemented, called DOT (deterministic oriented test pattern generation). It is based on a generalisation of the PODEM algorithm [9]. In the following, the main steps of the new method are explained.

Initial objective. In the PODEM algorithm for stuck-at faults, excitation of the fault to be tested is the primary objective. This is carried out by finding a suitable assignment to primary inputs (PI) belonging to the cone of the fault location in a way that the fault would be activated. However, in order to target specific defects inside a gate every gate input (GI) must have some certain value to cover the fault. The main difference to the traditional PODEM algorithm is therefore, that the initial objective does not consist of a value assignment to a single line, but of several objectives. To detect faults between interconnects the initial objective is to set the two shorted nodes to different values (i.e. two different objectives can exist).

GI vector reordering. In a general case, several GI vectors in a defect table for a gate can detect the same fault. Thus, the following defect coverage based heuristics was implemented to rearrange the order of GI vectors for which to generate tests. At the point in the algorithm, where we have to select the next GI vector to be targeted, calculation of the number of undetected faults within the faulty gate is carried out. For this GI vectors detecting a defect uniquely are chosen, if no such vector exist the GI vector detecting the highest number of uncovered faults will be chosen. This heuristics is aimed at minimizing the number of deterministic tests to be generated by maximising the fault coverage of each test.

Proceedings of the European Test Symposium (ETS'05)
1530-1877/05 $20.00 © 2005 **IEEE**

GI vector's defect coverage computation can also be done by summing up the probabilities of each fault being tested by each GI vector. It will become especially useful, if a 100 % test coverage is not the main goal, but a limited number of test vectors while achieving a high probabilistic test efficiency.

Backtrace and backtrack. To fulfil the test generation objectives in fault activation and propagation, a so called backtracing process is applied. During backtrace, values are recursively assigned to preceding circuit lines along a single path until a primary input (PI) is reached. The desired value is assigned to the PI and the circuit is simulated with the set of current value assignments. If the simulation gives inconsistent values to GI lines of the faulty gate then a *backtrack* is performed: the last PI assignment is inverted and circuit is simulated again. Otherwise, the next objective will be backtraced. The algorithm includes two types of objectives to backtrace. The first one is the initial objective explained above. The latter is the objective to propagate fault-effect D from the D-frontier towards the primary outputs (PO).

Fault effect propagation. When the initial objective has been satisfied by the backtrace process, the next step will be to propagate the fault from the faulty gate output to at least one of the primary outputs. To propagate the fault, further backtracing steps have eventually to be performed, each time followed again by a simulation. The possibility to propagate the fault from the D-frontier to at least one primary output with the current PI assignment has to be checked continuously by a process called X-path check. This procedure finds out whether there exists a signal path from the line with a fault effect (D) to some PO such that all the line values along the path hold don't-care (X) values. If X-path check fails for all the lines in the D-frontier then the PI assignment will be backtracked. The defect can be regarded as detected, if the fault is propagated to one or more primary outputs.

5. Experimental results

Table 2 presents the results of experiments carried out for detecting the AND-short defects in the 0.8µm CMOS technology. We used ISCAS85 circuits resynthesised by Synopsys Design Compiler. Two versions were synthesised: circuits with extension 'cmp' were created by a Design Compiler script that used the complex gates plus 2-input NOR and an inverter as the library basis. Circuits with extension 'all' are synthesised from a library containing all the gates for which the library defect tables were created.

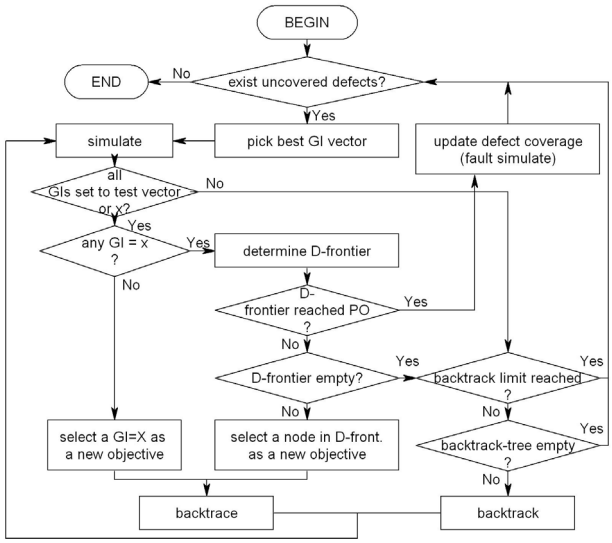

Figure 5. Defect-oriented ATPG algorithm (DOT)

The experiments were carried out using the new DOT algorithm described in the previous section and a conventional SAF based ATPG algorithm. Column 2 shows the total number of defects within the cells. Column 3 reflects the number of *gate redundant* defects. These are defects that cannot be covered by any gate input (GI) vector of a stand-alone gate. E.g. AND-short between the two inputs of the NAND gate is a gate redundant defect. In column 4, *circuit redundant* defects are counted. These are defects that cannot be tested as the circuit structure doesn't allow to generate a test for any of the GI vectors covering the defect. This redundancy is proved by the DOT algorithm. Column 5 shows the number of defects covered by DOT, while column 6 shows the result of SAF-ATPG. Column 7 gives the number of aborted defects that were not covered because of a given max. backtrack limit of 15000000 was exceeded.

Coverage measures can be calculated either using only the number of the defects (enumarative), or taking into account the probability of each defect. The defect coverage shows the ratio of detected defects versus the total set of defects, while the defect efficiency is the ratio of detected defects versus non-redundant defects. For the probabalistic coverage also the "gate level defect efficiency" is shown. Here, only the gate redundant defects are substracted,. This reveals the hierarchical approach used to determine redundancy in current approach. Note, that the probabilistic coverage should be considered since it takes into account the probabilities of occurrence of different defects.

Proceedings of the European Test Symposium (ETS'05)
1530-1877/05 $20.00 © 2005 **IEEE**

Table 2. Results of the defect-oriented ATPG (DOT) and a conventional SAF based ATPG tool

Circuit	# defects	# gate redundant defects	# Cct. redundant defects	# covered defects SAF-ATPG	# covered defects DOT	# aborted defects DOT	Enumerative				Probabilistic						# test pattern	
							Defect coverage DOT	Defect coverage SAF- ATPG	Defect efficiency DOT	Defect efficiency SAF- ATPG	Defect coverage DOT	Defect coverage SAF- ATPG	Gate level Defect efficiency DOT	Gate level Defect efficiency SAF- ATPG	Defect efficiency DOT	Defect efficiency SAF ATPG	DOT	SAF- ATPG
C432_cmp	1928	225	0	1694	1703	0	88,33%	87,86%	100,00%	99,47%	93,70%	93,18%	100,00%	99,48%	100,00%	99,48%	44	50
C499_cmp	4636	531	0	4105	4105	0	88,55%	88,55%	100,00%	100,00%	83,81%	83,81%	100,00%	100,00%	100,00%	100,00%	80	80
C880_cmp	3967	418	2	3543	3547	0	89,41%	89,31%	100,00%	99,89%	94,58%	94,41%	99,93%	99,77%	100,00%	99,84%	38	47
C1355_cmp	5360	520	38	4802	4802	0	89,59%	89,59%	100,00%	100,00%	89,82%	89,82%	99,10%	99,10%	100,00%	100,00%	83	85
C1908_cmp	3448	332	75	3040	3041	0	88,20%	88,17%	100,00%	99,97%	92,16%	92,16%	97,03%	97,03%	100,00%	99,99%	38	43
C2670_cmp	6746	721	55	5945	5967	3	88,45%	88,13%	99,95%	99,58%	85,88%	85,46%	98,71%	98,29%	99,95%	98,29%	102	110
C3540_cmp	9045	906	125	8004	8014	0	88,60%	88,49%	100,00%	99,88%	93,38%	93,19%	98,02%	97,84%	100,00%	99,81%	126	127
C5315_cmp	16034	1479	244	14305	14311	0	89,25%	89,22%	100,00%	99,96%	90,46%	90,39%	97,80%	97,73%	100,00%	99,93%	76	97
C6288_cmp	26994	3071	19	23904	23904	0	88,55%	88,55%	100,00%	100,00%	98,66%	98,66%	99,90%	99,90%	100,00%	100,00%	43	43
C432_all	1519	226	0	1278	1293	0	85,12%	84,13%	100,00%	99,05%	79,56%	78,61%	100,00%	99,05%	100,00%	99,05%	74	54
C499_all	4437	423	116	3898	3898	0	87,85%	87,85%	100,00%	100,00%	86,44%	86,44%	96,57%	96,57%	100,00%	100,00%	79	79
C880_all	3380	499	5	2868	2876	0	85,09%	84,85%	100,00%	99,66%	75,34%	75,00%	99,83%	99,50%	100,00%	99,66%	56	39
C1355_all	5302	641	22	4639	4639	0	87,50%	87,50%	100,00%	100,00%	85,27%	85,27%	99,46%	99,46%	100,00%	100,00%	87	87
C1908_all	3209	409	11	2789	2789	0	86,91%	86,91%	100,00%	100,00%	80,63%	80,63%	99,49%	99,49%	100,00%	100,00%	38	41
C2670_all	6090	703	61	5295	5326	0	87,45%	86,95%	100,00%	99,44%	79,69%	79,13%	98,54%	97,97%	100,00%	99,44%	107	110
C3540_all	7660	985	74	6584	6599	2	86,15%	85,95%	99,97%	99,76%	80,26%	80,05%	98,73%	98,52%	99,97%	99,76%	118	131
C5315_all	14794	1546	260	12988	12988	0	87,79%	87,79%	100,00%	100,00%	82,39%	82,39%	97,53%	97,53%	100,00%	100,00%	83	90
C6288_all	24433	4005	41	20387	20387	0	83,44%	83,44%	100,00%	100,00%	77,02%	77,02%	99,81%	99,81%	100,00%	100,00%	36	36

6. Conclusions

A new deterministic defect-oriented test generation method has been developed and implemented as a tool. In this method, we use a new functional fault model for mapping physical defects from transistor level to logic level. In the method, the conditional probabilities of defects can also be used for probabilistic defect coverage calculation. A probabilistic analysis of CMOS physical defects in the library cells has been carried out to create for each library cell a defect table. Based on these data, and by using the implemented deterministic ATPG tool (DOT), it was possible to prove the redundancy of physical defects and to generate deterministic test patterns to reach 100% defect efficiency (i.e. defect coverage in relation to nonredundant defects). A very low physical defect coverage (around 80-90%) was shown for 100% SAF test sets generated by a conventional ATPG tool. The defect efficiency of these test was, however, found as about 99% (in 50% of cases even 100%), but only after proving the redundancy of not covered defects by the implemented DOT tool. Moreover, it was shown, that the new algorithm achieves defect efficiency up to 1.6% higher than the conventional SAF ATPG tool.

Acknowledgments

This work has been supported by the EU projects REASON, eVIKINGSII, Estonian Science Foundation grants 5637, 5649, 5910, and Polish State Committee for Scientific Research project No. 4 T11B 023 24.

7. References

[1] J.M. Soden, C.F. Hawkins. Quality Testing Requires Quality Thinking. *Proc. Int. Test Conference*, 1993,pp. 596.

[2] W.Maly, et al. System Characterization of Physical Defects for Fault Analysis of MOS IC Cells. *Proc. 1984 ITC*, pp. 390-399, Philadelphia, 1984.

[3] J.P.Shen, W.Maly, J.Ferguson. Inductive Fault Analysis of MOS ICs. *IEEE Design&Test*, pp.13-26, 1985.

[4] P.Nigh, W.Maly. Layout - Driven Test Generation. *Proc.1989 ICCAD*, pp. 154-157, 1989.

[5] M.Jacomet, W.Guggenbuhl. Layout-Dependent Fault Analysis and Test Synthesis for CMOS Circuits. *IEEE Trans. on CAD*, vol. 12, pp. 888-899, 1993.

[6] R.Ubar, W.Kuzmicz, W.Pleskacz, J.Raik. Defect-Oriented Fault Simulation and Test Generation in Digital Circuits. *Proc. ISQED*, 2001, pp.365-371.

[7] M.Blyzniuk, W.Pleskacz, M.Lobur, W.Kuzmicz Estim. of Probability of Different Functional Faults Caused by Spot Defects in VLSI Circuits. Proc. TCSET 2000 Conf., 2000.

[8] W. A. Pleskacz, D. Kasprowicz, T. Oleszczak, W.Kuzmicz. CMOS Standard Cells Characterization for Defect Based Testing. Proc. DFT'01, pp.384-392, Oct. 2001.

[9] P.Goel. An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits. *IEEE Transactions on Computers*, vol. C-30, no.3 pp. 215-222, March 1981

[10] C.H: Stapper, I.M. Armstrong and K. Saji, "Integrated yield statistics" Proc. Of the IEEE, vol.71, no.4, pp 453-470, April 1983.

1530-1877/05 $20.00 © 2005 **IEEE**