ESTIMATE OF SIGNAL PROBABILITY
IN COMBINATIONAL LOGIC NETWORKS

S. Ercolani M. Favalli M. Damiani P. Olivo B. Riccó

DEIS, University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy

ABSTRACT

This work presents two new methods for the calculation of node signal probabilities in combinational networks that improve the state of the art by providing a better accuracy than existing algorithms and a deeper insight in the effects of first order correlations due to multiple fan-out reconvergencies. The proposed algorithms are shown to favorable compare with existing procedures in the analysis of significant benchmarks.

1. INTRODUCTION

The increase in complexity of modern ICs and the induced rise in the cost of test generation and fault simulation have shown the need for new approaches to these problems. Among the solutions proposed, *Pseudorandom Testing* is becoming progressively more important [1], mainly because it is widely applied in BIST, where Autonomous Linear Feedback Shift Registers (ALFSR) directly implemented on the chip, can be used to generate pseudorandom test vectors.

The most important figure of merit of a pseudorandom test pattern is the number of vectors (*test length*) needed to achieve a desired level of *fault coverage* that can be calculated with expensive fault simulation or, preferably, estimated by means of probabilistic models [1,2,3]. In this context, the problem of test quality verification, namely the estimate of the expected fault coverage and the identification of the random-resistant faults, requires the knowledge of the circuit detectability profile [1], which can be obtained exactly by means of exhaustive, though unpracticable, fault simulations. Alternatively, it can be evaluated from the *node signal probability* $p(j)$ [4], namely the probability that the signal value of a given node will be a "1" under a random assignment of an input vector. The task of determining node signal probabilities is very difficult since it represents a #P-complete problem [5,6], possibly even harder than a NP-complete one. To this purpose a number of heuristic algorithms have been proposed [4,6-10], but the problem is still waiting for more complete and satisfactory solutions.

In this paper, we present two original methods to calculate signal probabilities in combinational networks: one, faster and conceptually simpler, represents an original improvement of the technique described in Ref. [6] and provides a significantly better accuracy; the other more accurate, is also able to provide useful information of great help in the testability analysis of complex circuits.

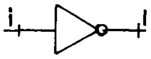
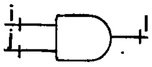
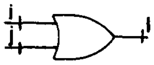
Our methods have been tested considering a few benchmark networks and using as reference for comparison exhaustive simulations providing exact signal probabilities or, when these were unpracticable, relevant Monte Carlo calculations. These analysis have shown that our methods, besides providing a deeper insight in the circuit behavior, are accurate and lead to good trade-offs between accuracy and computation time.

2. EXISTING METHODS FOR SIGNAL PROBABILITY COMPUTATION

The main difficulty in computing signal probabilities is due to the presence of reconvergent fan-out nodes (RFONs, namely those with fan-out > 1 and branches that reconverge within the circuit) that introduce functional dependencies and statistical correlations among the signals in the network.

For a tree-structured network (*i.e.* a circuit with no RFON), the signal probabilities can be evaluated in linear time according to the elementary rules of Table 1 [11], hereafter collectively referred to as *0-algorithm*. Furthermore, such rules can also be successfully applied to parts of general circuits that are fan-out free.

Table 1. Set of rules for calculating the node signal probabilities in absence of RFONs.

• Inverter gate  	$p(l) = 1 - p(i)$
• And gate  	$p(l) = p(i)p(j)$
• Or gate  	$p(l) = p(i) + p(j) - p(i)p(j)$

CH2696-3/89/0000/0132$01.00 © 1989 IEEE

In the general case, however, signal conditional probabilities whose order depends on multiple fan-out reconvergencies must be considered. To solve this problem, whose difficulty and complexity grow exponentially with the order of the considered conditional probability, several heuristic algorithms at different levels of accuracy have been proposed. Among these:

- the COP method [7], which assumes all signals to be independent (using the 0-algorithm) neglecting the effects of fan-out reconvergencies, is the simplest and fastest (linear in the number of gate N) but it is not accurate;
- the Weighted Averaging Algorithm (WAA) [6], which extends and improves the COP, attempts to take into account the first order effects of reconvergent fan-out. It is simple and linear in MN, where M is the number of inputs, but it is inaccurate in estimating the effects of RFONs internal to the circuit;
- the Cutting Algorithm (CTA) [4], which transforms a circuit with reconvergent fan-out into a tree structured equivalent one by means of arbitrary cuts critically determining the effectiveness of the method, leads to computation times that increase with the target accuracy;
- the STAFAN method [8], based on fault-free simulation, requires very long calculations to achieve a sufficient level of accuracy.

A detailed comparison about the accuracy and CPU requirements of these algorithms can be found in Ref. [10].

3. DYNAMIC WEIGHTED AVERAGING ALGORITHM

The Weighted Averaging Algorithm uses a weighted average of $p(j)$ values obtained by means of the 0-algorithm, setting each input probability alternatively at 1 and 0 while keeping the others at 0.5. Since it operates directly only on primary inputs, it is inaccurate in estimating the effects of RFONs internal to the circuit. In order to overcome this drawback, we developed an improved algorithm that applies the weighted averaging concept even to the internal RFONs.

3.1 The basic idea

Given a combinational network the signal probability $p(j)$ of a node j can be expressed by

$$p(j) = p(j/f_{=0})p(f_{=0}) + p(j/f_{=1})p(f_{=1}) \qquad (1)$$

where f is a RFON of the fan-in cone of j.

If f were the only RFON in the circuit, $p(j)$ could be exactly calculated by using eq. 1 and applying three times the 0-algorithm: setting the input probabilities to 0.5, first we evaluate $p(f)$, then, forcing the logical value of f to 0 and 1 we calculate the conditional probabilities. In the presence of more than one RFON, we obtain a value of $p(j)$ that is generally not exact and depends on the processed RFON. Since the actual value of $p(j)$ is influenced by other RFONs in the circuit, we estimate $p(j)$ for each RFON (hereafter referred to as $p_f(j)$).

In our approach, better estimates are obtained by *dynamically* updating $p(j)$ after having processed each RFON by weighted averaging all avaliable $p_f(j)$. In this way, since the accuracy of $p_f(j)$ depends on that of the values of all the inputs of th fan-out cone of f, at every step these are assigned taking into account all the information gathered from all processed RFONs. In this process, the order in which the RFONs are considered is of vital importance: the best way is to proceed from lower to higher level and within the same level from the RFON whose branches reconverge at the lower level.

3.2 The algorithm

The Dynamic Weighted Averaging Algorithm (DWAA), developed to implement the idea presented in the previous section, consists of the following main operations:

1) compile the circuit organizing it into levels and determine the order in which the RFONs are to be processed;
2) calculate the signal probabilities by means of the 0-algorithm. We will refer to these values as to $p(j,0)$, that is the probability of node j after having processed 0 RFON;
3) successively force each of the RFONs in the circuit to 1 and 0 and calculate the signal probabilities $p_f(j)$ of its fan-out cone from eq. 1, using as probability at the i^{th} input for its fan-out cone $p(i,t-1)$, namely the values calculated after having processed $(t-1)$ RFONs. After each step update the value of $p(j)$ as follows:

$$p(j,t) = \frac{p(j,t-1)ws(j,t-1) + p_f(j)w_f(j)}{ws(j,t-1) + w_f(j)} \qquad (2)$$

where the weighting factor

$$w_f(j) = abs(p_f(j) - p(j,0)) \qquad (3)$$

gives a measure of the deviation on $p(j)$ due to the influence of the considered RFON, and

$$ws(j,t-1) = \sum_{k=1}^{t-1} w_k(j) \qquad (4)$$

represents the sum of all w_f on all the $(t-1)$ RFONs already processed.

The operation # 3 stops when t equals the number of RFONs.

3.3 An example

As an example of the application of DWAA, we estimate the node signal probabilities of the simple circuit of Fig. 1, having two RFONs $i2$ and $n3$.

- Calculation of the signal probabilities $p(j,0)$ by means of the 0-algorithms.

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Ass.	.50	.50	.50	.50	.50	.50	–	–	–	–	–	–	–
Calc.	–	–	–	–	–	–	.25	.25	.75	.437	.375	.328	.58

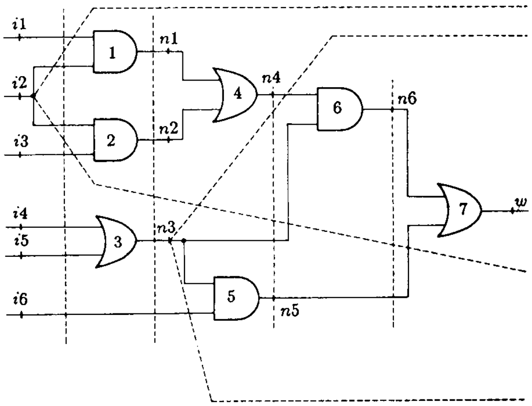


Figure 1.

- Force $i2$ to 0 and 1 and calculate $p_{i2}(j)$ by means of eq. 1, where j belongs to the fan-out cone of $i2$.

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Ass.	.50	.50	.50	–	–	–	–	–	.75	–	.375	–	–
Calc.	–	–	–	–	–	–	.25	.25	–	.375	–	.281	.55

Update $p(j,1)$ according to eq. 2. The values not explicitely updated keep the value of the previous step.

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Update	–	–	–	–	–	–	.25	.25	–	.375	–	.281	.55

- Repeat the previous step for $n3$, calculating $p_{n2}(j)$

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Ass.	–	–	–	–	–	.50	–	–	.75	.375	–	–	–
Calc.	–	–	–	–	–	–	–	–	–	–	.375	.281	.515

and $p(j,2)$

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Update	–	–	–	–	–	–	–	–	–	–	.375	.281	.526

The final estimation and the exact values of signal probability are summarized in the following table:

	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	w
Est.	.50	.50	.50	.50	.50	.50	.25	.25	.75	.375	.375	.281	.526
Exact	.50	.50	.50	.50	.50	.50	.25	.25	.75	.375	.375	.281	.515

4. CORRELATION COEFFICIENT METHOD

As already mentioned, the presence of fan-out nodes introduces correlation among the circuit signals. To deal with this problem, we have developed a method that explicitly accounts for first order conditional probabilities introducing *correlation coefficients* between couples of signals.

4.1 Correlation coefficients

From probabilistic theory, given two events A and B , the probability $P(AB)$ of $AB = A \cap B$ is given by $P(AB) = P(A/B)P(B)$ where $p(A/B)$ represents the probability of A given B [12].

In this paper we express this probability by means of correlation coefficients $C_{A,B}$ defined by the equation

$$P(AB) = P(A)P(B)C_{A,B} \tag{5}$$

hence

$$C_{A,B} = \frac{P(AB)}{P(A)P(B)} = \frac{P(A/B))}{P(A)} = \frac{P(B/A)}{P(B)} \tag{6}$$

If A and B are independent events $P(AB) = P(A)P(B)$, thus $C_{A,B} = 1$.

In the particular case of signal probability in a combinational network, an event is a node at 1, hence with the notation used in the previous sections $P(A) = P(j = 1) = p(j)$ and

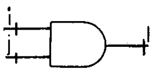
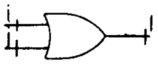
$$C_{i,j} = C_{j,i} = \frac{p(ij)}{p(i)p(j)} = \frac{p(i/j)}{p(i)} = \frac{p(j/i)}{p(j)} \tag{7}$$

$$p(i/j) = p(i)C_{i,j} \tag{8}$$

4.2 The main rules

Given a gate with two inputs (i, j), we can calculate exactly the probability $p(l)$ at the output by means of $p(i)$, $p(j)$ and their correlation coefficient $C_{i,j}$ as shown in Table 2.

Table 2. Set of rules for calculating the node signal probabilities through the correlation coefficients.

• And gate 	$p(l) = p(i)p(j)C_{i,j}$ $0 \leq p(l) \leq 1$ $0 \leq C_{i,j} \leq \frac{1}{p(i)p(j)}$
• Or gate 	$p(l) = p(i) + p(j) - p(i)p(j)C_{i,j}$ $0 \leq p(l) \leq 1$ $\frac{p(i)+p(j)-1}{p(i)p(j)} \leq C_{i,j} \leq \frac{p(i)+p(j)}{p(i)p(j)}$

The correlation coefficients can be calculated analytically by means of the set of basic rules of Table 3. The simple example of Fig. 2 shows how the rules of Table 3 allow to calculate the correlation coefficients for all cases occurring in a combinational network.

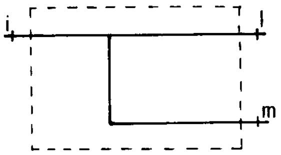
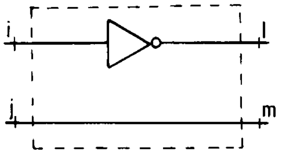
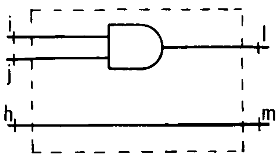
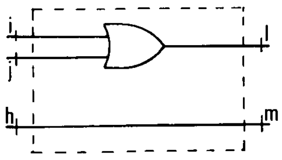
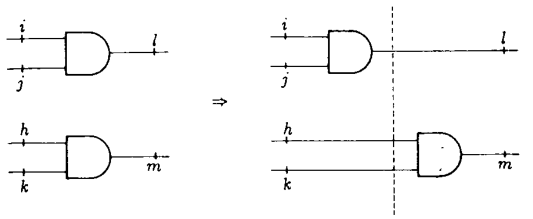
1) Primary inputs	Signals are assumed to be indipendent $p(i/j) = p(i)$, $p(j/i) = p(j)$ $\mathbf{C_{i,j} = 1}$
2) Fan-out node  	$i \equiv l \equiv m \qquad \Rightarrow p(l) = p(m) = p(i)$ $p(l/m) = 1$ $p(l)C_{l,m} = 1$ $\mathbf{C_{l,m}} = \dfrac{\mathbf{1}}{\mathbf{p(i)}}$
3) Not gate  	$l \equiv i^c \qquad \Rightarrow p(l) = 1 - p(i)$ $p(l/m) = 1 - p(i/j)$ $p(l)C_{l,m} = 1 - p(i)C_{i,j}$ $\mathbf{C_{l,m}} = \dfrac{\mathbf{1 - p(i)C_{i,j}}}{\mathbf{1 - p(i)}}$
4) And gate  	$l \equiv i \cap j \qquad \Rightarrow p(l) = p(i)p(j)C_{i,j}$ $p(l/m) = p(i/h)p(j/h)C_{i,j}$ $p(l)C_{l,m} = p(i)p(j)C_{i,h}C_{j,h}C_{i,j}$ $\mathbf{C_{l,m} = C_{i,h}C_{j,h}}$
5) Or gate  	$l \equiv i \cup j \qquad \Rightarrow p(l) = p(i) + p(j) - p(i)p(j)C_{i,j}$ $p(l/m) = p(i/h) + p(j/h) - p(i/h)p(j/h)C_{i,j}$ $p(l)C_{l,m} = p(i)C_{i,h} + p(j)C_{i,j} - p(i)p(j)C_{i,h}C_{j,h}C_{i,j}$ $\mathbf{C_{l,m}} = \dfrac{\mathbf{p(i)C_{i,h} + p(j)C_{j,h} - p(i)p(j)C_{i,h}C_{j,h}C_{i,j}}}{\mathbf{p(i) + p(j) - p(i)p(j)C_{i,j}}}$

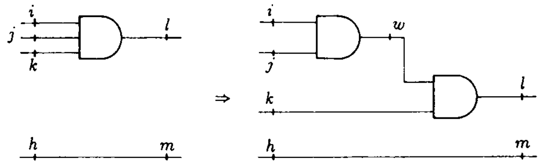
Table 3. Set of basic rules for determining the output correlation coefficients of a couple of signals given the input correlation coefficients and the input probabilities.



$$C_{l,h} = C_{i,h}C_{j,h}$$
$$C_{l,k} = C_{i,k}C_{j,k}$$
$$C_{l,m} = C_{l,h}C_{l,k} = C_{i,h}C_{j,h}C_{i,k}C_{j,k}$$

Figure 2.

The rules here given only for two input gates can be easily extended to the general case considering a multiple input gate as a cascade of two input ones. As an example, in Fig. 3 this procedure is applied to the case of a three input AND gate.



$$C_{w,h} = C_{i,h}C_{j,h}$$
$$C_{l,m} = C_{w,h}C_{h,k} = C_{i,h}C_{j,h}C_{h,k}$$

Figure 3.

In case 4 and 5 of Table 3 we assume

$$C_{ij,h} = C_{i,h}C_{j,h} \tag{9}$$

thus neglecting the dependencies of two signals to a third, hence the effect of second order conditional probabilies. The computation of second order correlation coefficients could in principle be done with a technique similar to that used to derive Table 3. This, however, would lead to an overwhelming increase of CPU time, for each level proportional to the cube of the number of gates at that level. On the other hand, the accuracy already achieved with only first order correlations (see later) suggests that an improvement in the cost/benefit trade off is very unlikely.

Neglecting higher order correlations, the calculated $C_{i,j}$ can lead to gate output probabilities outside the 0-1 interval. This can be avoided by limiting $C_{i,j}$ within the bounds given in Table2.

4.3 The algorithm

From the rules of Table 2 and 3, successively applied proceeding from primary inputs to outputs, we can calculate the node probability by means of the following algorithm:

1) compile the network transforming possible multiple input gates into a cascade of two input ones organizing the circuit into levels;
2) initialize the correlation coefficients and the probabilities at primary inputs. Generally in pseudorandom testing the input signals have probability = 0.5 and are considered independent,thus $C_{i,j} = 1$;
3) calculate the node probabilities and the signal correlation coefficients at each level by successively applying the rules of Table 2 and 3. Check that the calculated $C_{i,j}$ are within the bounds of Table 2. If not, assign them the nearest bound values.

To save computation time it can be useful to recognize during circuit compilation all the couples of independent signals and the gates for which there is no fan-out reconvergencies because for parts of the circuit that are fan-out free or tree-structured it is simpler and faster to use the 0-algorithm.

4.4 An example

As an example of the application of the Correlation Coefficient Method (CCM), we estimate the node signal probabilities of the 2-input multiplexer of Fig.4, where G4 is the only gate at which there is reconvergency.

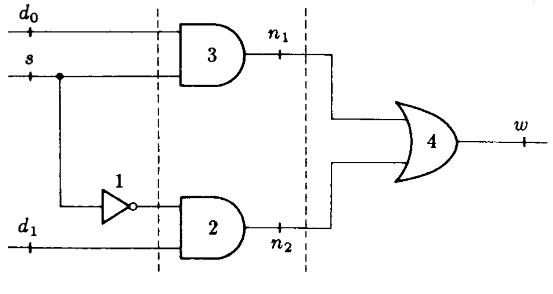


Figure 4.

- At level 0 initialize the input probabilities and the correlation coefficients assuming the signals to be independent. Introduce the correlation due to s:

  $p(d0) = p(d1) = p(s) = 0.5;$

  $$C_{d0,d1} = C_{d0,s} = C_{d1,s} = 1; \quad C_{s,s} = \frac{1}{p(s)} = 2;$$

- Calculate the node probabilities and the correlation coefficients at level 1 by means of the rules of Table 1, 2, 3:

  $p(ns) = 0.5;$

  $$C_{s,ns} = \frac{1 - p(s)C_{s,s}}{1 - p(s)} = 0;$$

- At the next level in the network repeat the previous step:

  $p(n1) = p(n2) = 0.25;$

  $C_{n1,n2} = C_{s,ns}C_{s,d1}C_{d0,ns}C_{d0,d1} = 0;$

- At last level calculate only the signal probabilities.

  $p(w) = 0.25 + 0.25 - 0.25 * 0.25 * 0 = 0.5;$

As can be easily verified by direct inspection the results are correct.

5. RESULTS AND DISCUSSION

The two new method described in the previous sections have been applied on a few commercial circuits and benchmarks suggested in [13]. The results have been compared with references obtained through exhaustive simulations or, when these were upracticable, with extensive Monte Carlo calculations (performed until achieving a satisfactory convergence in the signal probabilities). In order to measure the quality of our results, we define the error as the root mean square (RMS) deviation from the reference values on all the internal nodes. As this regard it is worth mentioning that other authors use different error definitions. For instance Ref. [10] considers the average on all the nodes of the absolute difference from the reference results. We prefer to use the RMS deviation because in this case each single error on each internal node is weighted depending on its distance from the exact value, however the results provided by our methods are significantly better than those of previous works regardeless of the actual error definition.

Table 4. The reported values represent the RMS and the ME from the reference results obtained with exhaustive (†) and Monte Carlo (*) simulations.

Circuit Name	COP	WAA	DWAA	CCM
SN74LS157†	0.0072 0.0156	0.0072 0.0156	0.0072 0.0156	0.0000 0.0000
SN74LS181†	0.0519 0.1715	0.0342 0.1539	0.0335 0.1252	0.0151 0.0730
C432*	0.0925 0.3512	0.0829 0.3450	0.0614 0.3052	0.0612 0.3441
C499*	0.0026 0.0224	0.0026 0.0224	0.0023 0.0224	0.0003 0.0008
C880*	0.0314 0.0722	0.0195 0.0618	0.0164 0.0550	0.0061 0.0524
C1355*	0.0900 0.1475	0.0747 0.1474	0.0595 0.1305	0.0208 0.0473

In Table 4 we compare the RMS values and the maximum error (ME) calculated with our methods (DWAA, CCM) with those given by some of the procedures mentioned in section 2, namely COP and WAA. As can be seen, both the RMS and the ME calculated with our two algorithms are significantly smaller than those achieved with the others. In particular the DWAA gives better result than the WAA increasing the the number of RFONs and the size of the circuit. The WAA is more powerful only for those circuits having the RFONs close to the primary inputs.

From Table 4 it can also be seen that the results obtained on SN74LS157 (data selector) confirms that the CCM is exact for circuits presenting only first-order conditional probability.

Table 5.

	Time Requirement
COP	N
WAA	$(1 + 2M)N$
DWAA	$\leq (1 + 2F)N$
CCM	$\leq \sum_L N_L(N_L - 1)/2$

Table 5 shows a comparison of the same methods of Table 4 in terms of time requirement. Instead of giving the actual CPU time, strictly depending on the the algorithm implementation and on the computer architecture, we prefer to concentrate on the relationship between calculation time and circuit parameters. The DWAA is linear in the product FN, where F if the number of RFONs. In particular, since for each RFON we have to consider only that part of the network that is in its fan-out cone, $(1+2F)N$ is the maximum time consuption (for a circuit in which every RFON affects all the gates).

The CCM is linear in the number of levels (L) and pseudo-quadratic in that of the gates per level (N_L). Its performances depend on the internal structure of the network and not only on the number of gates: for a square circuit, namely when we can assume $N_L = L = \sqrt{N}$ the complexity of the CCM becomes $N(\sqrt{N} - 1)/2$. In general the calculation time is less than the limit indicated in Table 5 because the CCM calculates the correlation coefficients only for couples of signals that are dependent from each other.

References

[1] K.D. Wagner, C.K. Chin and E.J. McCluskey, "Pseudorandom testing", *IEEE Transactions on Computer*, **C-36**, 332 (1987)

[2] E.J. McCluskey, S. Makar, S. Mourad and K.D. Wagner, "Probability models for pseudorandom testing", *IEEE Transactions on CAD*, **7**, 68 (1988)

[3] W.K. Huang, M. Lightner and F. Lombardi, "Predicting fault coverage for random testing of combinational circuits", *Proceeding 1987 International Test Conference*, 834

[4] J. Savir, G.D. Ditlow and P.H. Bardell, "Random pattern testability", *IEEE Transaction on Computer*, C-**33**, 29 (1984)

[5] P. Goel, "Test generation costs and projections", *Proceeding* 12th *Annual Fault Tolerant Computer Symposium*, 101 (1982)

[6] B. Krishnamurthy and I.G. Tollis, "Improved Techniques for estimating signal probabilities", *Proceeding 1986 International Test Conference*, 244

[7] F. Brglez, P. Pownall and R. Hum, "Application of testability analysis: from ATPG to critical delay path tracing", *Proceeding 1984 International Test Conference*, 705

[8] S.K. Jain and V.D. Agrawal, "Statistical fault analysis", *IEEE Design & Test*, February 1985, 38

[9] H.J. Wunderlich, "Protest: a tool for probabilistic testability analysis", *Proceeding* 22nd *Design Automation Conference*, 204 (1985)

[10] P. Camurati, P.Prinetto and M. Sonza Reorda, "Random testability analysis: comparing and evaluating existing approaches", *Proceeding 1988 International Conference on Computer Design*

[11] K.P. Parker and E.J. McCluskey, "Probabilistic Treatment of General Combinational Networks", *IEEE Transaction on Computer*, C-**24**, 668 (1975)

[12] M. Loève, "Probability theory", Van Nostrand, 2nd Edition

[13] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran", *Proceeding 1985 International Symposium on Circuit And Systems*