MULTIPLE DISTRIBUTIONS FOR BIASED RANDOM TEST PATTERNS

Hans-Joachim Wunderlich

University of Karlsruhe
Institute of Computer Design and Fault-Tolerance
(Prof. Dr. D. Schmid)
Postfach 6980, D-7500 Karlsruhe 1
F. R. Germany

Abstract:

The test of integrated circuits by random patterns is very attractive, since no expensive test pattern generation is necessary and the test application can be done by a self-test technique or externally using linear feedback shift-registers. Unfortunately not all circuits are random-testable, since the fault coverage would be too low or the necessary test length would be too large. In many cases the random test lengths can be reduced by orders of magnitude using weighted random patterns. But there are also some circuits where no single optimal weight exists. In this paper it is shown that the problem is solved using several distributions instead of a single one. Furthermore an efficient procedure is presented computing the optimized input probabilities.

Thisway all combinational circuits can be made random-testable. Fault simulation with weighted patterns shows a complete coverage of all non-redundant faults. The patterns can be successively produced by an external chip, and an optimized test scheme for circuits in a scan design can be established.

As a result of its own formulas are derived determining sharp bounds of the probability that all faults are detected.

Keywords: Random tests, biased random patterns, multiple distributions, low cost test.

1) Introduction

Testing by random patterns has many advantages compared to other test strategies, for instance the self test capability, less computing time and the high coverage of parametric faults. An extensive literature has been published during the last few years concerning problems of random tests, as computing fault detection probabilities and test lengths. Unfortunately most of these papers are only dealing with special views of the subject.

Therefore it is necessary to list some basic facts as prerequisites of the following investigations. Among these basic facts, there is a new theorem establishing a sharp bound of the probability that all faults of a given set are detected by a given amount of random patterns. Another theorem proves that a real random test and a pseudo-random test by shift-register sequences require the same length, if the number of primary inputs is sufficiently large.

It is shown that the fault coverage increases, and the overall testlength decreases, if several random pattern sets with different distributions are applied. The optimized input probabilities can be computed numerically, if a procedure is available, estimating fault detection probabilities, and satifying certain restrictions. The restrictions are discussed at the end of section 2.

In section 3 the complexity of computing an optimized random test scheme is determined, and since this problem ist at least np-hard, we avoid the exact calculation using an efficient heuristic in section 4. Some implementation details are given in section 5, and results are discussed in section 6. Finally we present a system· generating weighted random test patterns according to multiple distributions, which is used for the external test of circuits with an integrated scan-path.

2) Basic facts

2.1 Fault detection probabilities

One of the main concepts of random tests is the computation of fault detection probabilities. Many tools and algorithms have been proposed during the past years estimating these probabilities (e.g. [BDS83], [AgJa84], [Wu85], [ChHu86], [AaMe87]). Most of them are restricted to the usual stuck-at fault-model, but an extension to more complex faults is possible in a straightforward manner, unless a sequential behavior is is involved [WuRo86]. The precision of these tools, however, is limited by the following fact:

Fact 1: Computing fault detection probabilities is at least np-hard.

This is a simple consequence of the np-completeness of the fault detection problem [IbSa75]. The mentioned fact has led some people to the conjecture that a stochastical Monte Carlo algorithm would yield a higher precision. But this is not true:

Fact 2: Estimating fault detection probabilities is #-complete, that is, one cannot expect a stochastical algorithm with a sample size bounded by a polynomial in the reciprocal of the relative estimation error.

This result is derived using elementary concepts of complexity theory found in [GaJo79]. In a more formal way, the pattern number N may exponentially increase with the reciprocals of the maximal relative estimation error (ε) and of the probability (P) that this error is not exceeded (δ), where f is the real detection probability and f ' is an estimated value provided by a sample size N:

(1) $$P\left(\frac{|f - f'|}{f} \geq \varepsilon\right) \leq \delta$$

· This research has been supported by the BMFT (Bundesministerium für Forschung und Technologie) of the Federal Republic of Germany under grant NT 2809 A 3

Both facts point out that we cannot expect tools estimating fault detection probabilities with arbitrary high precision neither analytically nor stochastically. This has consequences on the numerical stability of algorithms computing optimal distributions, and it will be discussed in section 2.3. The intrinsic error also makes useless algorithms computing random test lengths in a very sophisticated way, if the input is based on estimated fault detection probabilities.

2.2 Fault detection probabilities and test lengths

Often it is discussed that the *pseudo*-random property has to be considered, and there are some papers published on this topic [WAGN87]. But here we have the following fact:

Fact 3: For realistic circuits the difference between the length of a random test and the length of a pseudo-random test is negligible.

The fact is an immediate consequence of theorem 1. It holds for circuits with a realistic number of primary inputs, where all possible input patterns cannot be enumerated exhaustively. Only in this case a random test makes sense, and the random pattern set will be a very small part of an exhaustive test set.

Theorem 1:

Let p be the detection probability of a fault f in a combinational circuit with i inputs, and let ε be the escape probability that f is neither detected by N_r random patterns nor by N_p pseudo-random patterns. For $2^{i/2} \geq N_r$ we have $N_r = N_p$.

Proof:

Fault detection by random patterns follows the binomial distribution, and we have $\varepsilon = (1-p)^{N_r}$ or $\ln(\varepsilon) = N_r \ln(1-p)$. Expressing the exponention function by its power series we have

(2)

$$\ln(\varepsilon) = N_r \ln\left(e^{-p} - \sum_{i=2}^{\infty}\frac{(-p)^i}{i!}\right) = N_r \ln\left(e^{-p} - p^2\sum_{i=0}^{\infty}\frac{(-p)^i}{(i+2)!}\right)$$

$$= N_r\left(-p + \ln\left(1 + \frac{p^2}{e^{-p}}\sum_{i=0}^{\infty}\frac{(-p)^i}{(i+2)!}\right)\right).$$

Straightforward computation shows

(3)
$$\frac{\sum_{i=0}^{\infty}\frac{(-p)^i}{(i+2)!}}{e^{-p}} < \frac{1}{2}$$

and by iteration we can prove

(4)
$$N_r p \leq -\ln(\varepsilon) \leq N_r p + N_r p^2.$$

Fault detection by pseudo-random patterns follows hypergeometric distribution, that is

(5)
$$\varepsilon = \frac{\binom{2^i - 2^i p}{N_p}}{\binom{2^i}{N_p}} = \frac{(2^i(1-p))!\,(2^i - N_p)!}{(2^i(1-p) - N_p)!\,2^i!} = \prod_{k=0}^{N_p-1}\frac{2^i(1-p) - k}{2^i - k}$$

$$= \prod_{k=0}^{N_p-1}\left(1 - \frac{2^i}{2^i - k}p\right).$$

Using the same argument as before we have

(6)
$$\sum_{k=0}^{N_p-1}\frac{2^i p}{2^i - k} \leq -\ln(\varepsilon) \leq \sum_{k=0}^{N_p-1}\left(\frac{2^i p}{2^i - k} + \left(\frac{2^i p}{2^i - k}\right)^2\right)$$

or

(7)
$$2^i p \sum_{k=2^i-N_p+1}^{2^i}\frac{1}{k} \leq -\ln(\varepsilon) \leq 2^i p \sum_{k=2^i-N_p+1}^{2^i}\left(\frac{1}{k} + \frac{2^i p}{k^2}\right).$$

Hence

(8)
$$2^i p N_p \frac{1}{2^i} \leq -\ln(\varepsilon) \leq \frac{2^i p \cdot N_p}{(2^i - N_p + 1)}\left(1 + \frac{2^i p}{2^i - N_p + 1}\right)$$

and since

(9)
$$2^i N_p \frac{1}{2^i - N_p + 1} = N_p\left(1 + \frac{N_p - 1}{2^i - N_p + 1}\right) =$$

$$= N_p + \frac{N_p^2 - N_p}{2^i - N_p + 1} \leq N_p + \frac{2^i - N_p}{2^i - N_p + 1} < N_p + 1$$

we have

(10)
$$pN_p \leq -\ln(\varepsilon) < p(N_p + 1)\left(1 + \frac{N_p + 1}{N_p}p\right)$$

$$< p(N_p + 1) + 2p^2(N_p + 1).$$

Since for usually small detection probabilities p the terms $2p^2(N_p+1)$ and $p^2 N_r$ both are smaller than p, and since N denotes integer numbers, the inequations (4) and (10) lead to $pN_p \approx -\ln(\varepsilon)$, $pN_r \approx -\ln(\varepsilon)$, and thus $N_p = N_r$.

qed.

As a consequence we can use the random assumption without any loss of generality for those circuits where an exhaustive test is impossible. For instance, if we have to apply less than 8000 patterns to a circuit with more than 25 primary inputs, then random and pseudo-random pattern sets will exactly have the same size.

Let now F be a set of faults of the combinational circuit C with inputs I, with the only restriction that no sequential behavior is induced. Let $X := \langle x_1,\dots,x_n\rangle \in [0,1]^I$ be a tupel of real numbers, one number for each primary input. These input probabilities determine the probability of being "1" for each primary input, and for each fault they determine its detection probability $p_f(X)$. The probability that each single fault of F is detected by N random patterns is often estimated by the formula

(11)
$$J_N(X) = \prod_{f\in F}\left(1 - (1 - p_f(X))^N\right).$$

Of course formula (11) only holds if we assume that the detection of some faults by N patterns forms completely independent events. Therefore some authors try to compute an exact value by means of Markov-theory [BaSa83], but the next theorem shows that formula (11) indeed is a very precise estimation.

Let $\langle f_i\rangle_{i\leq\mu}$ be an enumeration of F, where $i < j$ implies $p_{fi} \leq p_{fj}$. In order to simplify the notation we omit the concrete distribution X. The expression P(A,N) denotes the probability to detect all faults of the set A by N random patterns. Then we can show:

Theorem 2:

Set
$$J_N := \prod_{f\in F}(1-(1-p_f)^N).$$
Then
$$J_N - (1-J_N)\sum_{j=2}^{\mu}(1-p_{f_j})^N \le$$
$$\le P(F,N) \le$$
$$\le J_N + \sum_{j=2}^{\mu}(1-p_{f_j})^N\prod_{k=1}^{j-1}(1-(1-p_{f_k})^N)$$

Proof:

We set $\delta_{n+1} := P(\{f_i | i \le n+1\}, N) - \prod_{i=1}^{n+1}(1-(1-p_{f_i})^N)$.

Now we have $P(F,N) = J_N+\delta_\mu$, and using the Bayesian formula we can estimate

$$\delta_{n+1} = P(\{f_i | i \le n\}, N\} - \prod_{i=1}^{n+1}(1-(1-p_{f_i}))^N$$
$$- (1-p_{f_{n+1}})^N P(\{f_i | i \le n\}, N \mid \text{no pattern detects } f_{n+1})$$
$$= P(\{f_i | i \le n\}, N) - (1-(1-p_{f_{n+1}})^N)\prod_{i=1}^{n}(1-(1-p_{f_i})^N)$$
$$- (1-p_{f_{n+1}})^N P(\{f_i | i \le n\}, N \mid \text{no pattern detects } f_{n+1})$$
$$= \delta_n + (1-p_{f_{n+1}})^N\Big(\prod_{i=1}^{n}(1-(1-p_{f_i})^N) -$$
$$P(\{f_i | i \le n\}, N \mid \text{no pattern detects } f_{n+1}).$$

Thus $\delta_{n+1} \le \delta_n + (1-p_{f_{n+1}})^N\prod_{i=1}^{n}(1-(1-p_{f_i})^N)$,

and since $d_1 = 0$:
$$\delta_{n+1} \le \sum_{j=2}^{n+1}(1-p_{f_{j+1}})^N\prod_{k=1}^{j-1}(1-(1-p_{f_k})^N).$$

On the other hand $P(\{f_i | i \le n\}, N \mid \text{no pattern detects } f_{n+1}) \le 1$, and we have

$$\delta_{n+1} \ge \delta_n + (1-p_{f_{n+1}})^N\Big(\prod_{i=1}^{n}(1-(1-p_{f_i})^N) - 1\Big) \ge$$
$$\ge \delta_n - (1-p_{f_{n+1}})^N(1-J_n) \ge -(1-J_N)\sum_{j=2}^{n+1}(1-p_{f_j})^N.$$

qed.

Using this theorem we can state

Fact 4: Let $J_N \approx 1$ be the derived probability to detect all faults. Formula (11) underestimates less than $O(|\ln(J_N)|)$ and overestimates less than $O(|(1-J_N)\cdot\ln(J_N)|)$.

For instance if we have 3 faults with $p_{f1} = 10^{-7}$, $p_{f2} = 5\cdot10^{-7}$ and $p_{f3} = 10^{-6}$ then using formula (11) we would need $N = 69\cdot10^6$ patterns in order to detect all faults with probability 0.999. The estimation of theorem 2 yields

$$0.999 - 10^{-18} \le P(\{f_1, f_2, f_3\},N) \le 0.999 + 10^{-15}.$$

The following fact has already been observed in [BaSa83]:

Fact 5: Only the few faults with lowest detection probability have impact on the necessary test length.

In [Wu87a] it is discussed that those faults can be neglected which have a detection probability more than 10 times larger than the minimal detection probability.

The next statement has already been established in [Shed77]:

Fact 6: The necessary number of random patterns increases linearly with the reciprocal of the minimal fault detection probability.

Thus during a conventional random test the size of a test set can grow exponentially with the number of inputs. For instance consider an AND32 (figure 1) where each input is set to "1" with probability x.

Then an arbitrary stuck-at-0 fault is detected with probability x^{32}, and each of the 32 stuck-at-1 faults with probability $(1-x)\cdot x^{31}$. For x = 0.5 and test confidence 0.999 formula (11) yields

$$0.999 = (1-(1-2^{-32})^N)^{33},$$

and $N \approx 4.48\cdot10^{10}$.

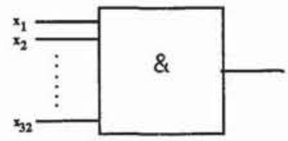

Figure 1: 32 Input AND

But using unequiprobable patterns, i.e. $x \ne 0.5$, test lengths can be reduced drastically ([Wu85], [LBGG86]). For example setting

$$x := \sqrt[32]{0.5},$$

we would need approximately $N \approx 6\cdot10^2$ patterns.

Fact 7: Using unequiprobable random patterns the test lengths can be reduced.

In [Wu87a,b] an efficient procedure computing optimized input probabilities was presented. But unfortunately some circuits are resistant to optimizing. For the connection of an AND32 and an OR32 in figure 2 no solution better than x = 0.5 exists.

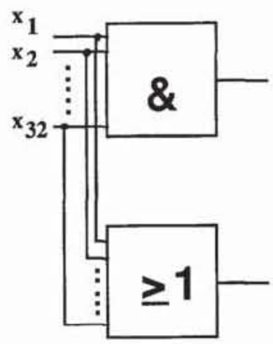

Figure 2: Not random-testable circuit

This problem is solved by firstly applying 600 patterns with input probability

$$x := \sqrt[32]{0.5},$$

and then 600 patterns with input probability

$$x := 1 - \sqrt[32]{0.5},$$

For the rest of, this paper we are dealing with the problem to compute several distributions X for random patterns in order to minimize the overall test length, based on tools estimating fault detection probabilities

2.3 Efficiency and accurcacy of the testabillity measures

Computing optimized distributions X is essentially done by numerical algorithms maximizing formula (11). The involved fault detection probability $p_f(X)$ has to be determined by a testability measure meeting the following requirements:

a) High Efficiency:

During optimization the detection probabilities p_f have to be evaluated very often for different arguments. Due to the already mentioned inherent problem complexity this is not possible by an exact computation. Recently several algorithms have been proposed, exactly computing fault detection probabilites using a 4-valued logic [ChHu86] or using some graph-theoretic properties of the circuit [SETH86]. But until now no reports about their measured performance are available. Therefore we have to use heuristics estimating fault detection probabilities, and dispense with the exact computation.

b) Unique results:

The so-called cutting algorithm is a heuristic computing bounds of the detection and signal probabilities [BDS83]. If one of the bounds is constant 0 or 1, this information is not sufficient, since especially the faults with low detection probabilities are interesting. Thus we demand a real number as an estimation.

c) Handling weighted input probabilities:

For each fault its detection probability will be computed several times for different input probabilites X. If the input probabilities differ only in few positions, the algorithm should take advantage of this fact.

d) No random errors:

The algorithm runs for different input probabilities, and the estimation error should not be a random variable. If the error is random, we have to deal with a stochastical optimizing problem which has a very high complexity. Optimizing procedures like the Newton iteration in general do not converge based on stochastical derived inputs. Only in special cases these algorithms can be modified, for instance for PLAs as described in [Wu87d].

These four requirements are fulfilled by the tool PROTEST (Probabilistic testability analysis) as described in [Wu85], [Wu87a]. Furthermore this tool has the advantage that the user can control the trade-off between the precision of the estimation and the required computing time. All results reported in this paper are provided by PROTEST.

3) Optimizing input probabilities

Now we can try to formulate the optimizing problem in a more formal way:

Problem A:

Let G be the desired probability to detect all faults. Find a number k, k distributions X^i, and k numbers N_i, $i = 1,\ldots,k$, such that

$$G \le \prod_{f\in F}\Big(1 - \prod_{i=1}^{k}(1 - p_f(X^i))^{N_i}\Big), \text{ and } N := \sum_{i=1}^{k} N_i \text{ is minimal.}$$

The problem is solved, if we set k equal to the minimal number of deterministic test patterns, that is the size of the smallest possible test set. Then each $X^i \in [0,1]^n$ represents a test pattern, we have $N_i = 1$ for each pattern, and $N = k$. But the problem to find a minimal test set has been proven to be np-complete [AkKr84], hence there is no hope to develop an efficient CAD tool based on a solution for problem A. Even the weaker problem B will turn out to be np-hard:

Problem B:

Let G and k be given. Find k distributions X^i and numbers N_i, $i = 1,\ldots,k$, such that

$$G \le \prod_{f\in F}\Big(1 - \prod_{i=1}^{k}(1 - p_f(X^i))^{N_i}\Big), \text{ and } N := \sum_{i=1}^{k} N_i \text{ is minimal.}$$

A single moment of consideration proofs that problem A can be reduced to problem B, and an efficient algorithm cannot be expected either. Therefore our goal is not an optimal solution of problem A or B, but we are content to find an efficient optimizing procedure. Figure 2 indicates that optimizing input probabilities can be prevented by contradictory requirements of some faults. Hence we formulate our problem as follows:

Optimizing problem:

Let G and k be given. We are searching a partition $<F_1,\ldots,F_k>$ of $F := F_1 \cup \ldots \cup F_k$, distributions $X^1,\ldots,X^k$ and numbers $N_1,\ldots,N_k$, such that

(12)
$$G \le \prod_{i=1}^{k}\prod_{f\in F_i}\Big(1 - (1 - p_f(X^i))^{N_i}\Big), \text{ and } N := \sum_{i=1}^{k} N_i$$

is sufficiently small.

For k := 1 this problem has already been solved in [Wu87a,b], and we now list some basic results of this paper. For the input probabilities $X := <x_1,\ldots,x_n> \in [0,1]^n$ we have for all faults f

(13)

$$p_f(X) = p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n) + x_i(p_f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) - p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)).$$

This is a straightforward consequence of Shannon's formula. Now we can compute a fault detection probability and its partial derivative for an arbitrary value of x_i, if we know the values under the conditions that input i is constant "0" and constant "1":

(14)
$$\frac{dp_f(X)}{dx_i} = p_f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) - p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n).$$

By some straightforward approximations formula (12) leads to

(15)
$$\ln(G) \approx \sum_{f\in F} -(1 - p_f(X))^N \approx -\sum_{f\in F} e^{-p_f(X)N}$$

We call a tupel $X \in [0,1]^n$ *optimal*, if the objective function

(16)
$$\delta_N^F(X) := \sum_{f\in F} e^{-p_f(X)N}$$

is *minimal*. Obviously this corresponds to the fact that the probability to detect all faults by N patterns is maximal.

Minimizing the objective function would need exponential effort in general. But a sufficient heuristic is found, since the first partial derivative of the objective function can be computed explicitly:

(17)
$$\frac{d\delta_N^F(X)}{dx_i} = -\sum_{f\in F} N(p_f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) - p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n))e^{-p_f(X)}.$$

The next step shows that the second derivative is positive everywhere:

(18)
$$\frac{d^2\delta_N^F(X)}{dx_i^2} = -\sum_{f\in F} N^2(p_f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n) - p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n))^2 e^{-p_f(X)}.$$

Thus the objective function is strictly convex with respect to a single variable, and the explicit formula of (18) can be used to find the optimal value for x_i by the bisection method, the regula falsi or the Newton iteration. The complete optimizing procedure is:

```
Procedure Optimize
        (F: Faultsets, X: Startvector)
        Old := 2δN(X)
        New := δN(X).
        While  Old > (1+λ)New+ε do
               Old := New.
               For i := 1 to n do
               Search optimal value y
               for input i.
               xi := y
               New := δN(X)
```

The parameters λ and ε are specified by the user, and they determine another trade-off between accuracy and computing time. In the next sections we discuss the extension to multiple distributions.

4) Partioning of a fault set

In order to gain efficiency, the optimizing problem is solved by splitting the fault set into two subsets iteratively. In this section it is discussed, how to find two tupels $V_1, V_2 \in [0,1]^I$ and a partition $F_1 \cup F_2 = F$, such that the sum of the two corresponding objective functions is minimized:

$$\delta_N^{F_1}(V_1) + \delta_N^{F_2}(v_2) := \sum_{f\in F_1} e^{-p_f(V_1)N} + \sum_{f\in F_2} e^{-p_f(V_2)N} < \delta_n^F(X).$$

For each $F^* \subset F$ the objective function $\delta_N^{F^*}$ may be multimodal and its global minimization would need exponential effort. For this reason we do not try to compute a global minimum, but we are looking for a direction, where starting from a tupel X_0 the decrease of $\delta_N^{F^*}$ is maximal. The next theorem will give a helpful hint:

Theorem 3:

Let $U \subset R^n$ be convex, $\zeta : U \to R$, and
$$\mathrm{grad}(\zeta) := \left(\frac{d\zeta}{x_i}\right)_{1\le i\le n}$$
be the gradient of ζ. For each $X_0 \in U$ the vector $-\mathrm{grad}(\zeta)(X_0)$ indicates the direction of strongest decrease. If ζ is linear a local minimum is found on the line $X_0 - \alpha\mathrm{grad}(\zeta)(x_0)$, $\alpha \ge 0$.

Proof: Mathematical calculus.

Even though $\delta_N^{F^*}$ is not a linear function, theorem 3 claims that
$$-\mathrm{grad}(\delta_N^{F^*})(X_0)$$
is the required direction. Thus we define the new function
$$\xi_N^{F^*} : R^+ \cup \{0\} \longrightarrow R$$
by $$\xi_N^{F^*}(\alpha) := \delta_N^{F^*}(X_0 - \alpha\mathrm{grad}(\delta_N^{F^*})(X_0))$$

The formula

(19)
$$D(F^*, N, X_0, 0) := \frac{d\xi_N^{F^*}(\alpha)}{d\alpha}(0)$$

exactly measures the decrease of our objective function in its optimal direction. The solution of

(20) $$D(F^*,N,X_0,\gamma) = 0$$

provides input probabilities
$$X_0 - \gamma\mathrm{grad}(\delta_N^{F^*})(X_0)$$
defining a minimum point in this direction. Therefore our partitioning problem is solved by F_1 and F_2 such that

(21) $$D(F_1,N,X_0,0) + D(F_2,N,X_0,0) > 0$$

is maximal. It should be noted that for linear functions this proceeding would be optimal indeed.

For the rest of this section the tasks necessary for partitioning are discussed. These tasks have to be done only for the small subset of faults with lowest detection probability due to fact 5). If this set is small enough, the presented method will compute a global optimal solution maximizing formula (21). For large fault sets computing time can be saved, if the method is somewhat simplified.

a) Computing the gradient

The gradient
$$X_0 - \gamma\mathrm{grad}(\delta_N^{F^*})(X_0)$$
can be computed explicitly using formula (16). If additionally formula (13) is used, it is immediately seen, that we only have to compute $p_f(X)$ and either $p_f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$ or $p_f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)$ for this purpose.

b) Sorting the fault set:

For each fault let

$$d_f(X_0) := \sqrt{\sum_{i=1}^{n} \left(\frac{de^{-p_f(X)N}}{dx_i} \right)^2 (X_0)} = \left\| \mathrm{grad}\,(e^{-p_f(X)N})(X_0) \right\|$$

be the Euclidian norm of the gradient of

$$e^{-p_f(X)N}$$

in X_0, and let $< f_i >_{i \le k}$ be an enumeration of F with

$$i \le k \Rightarrow d_{f_i}(X_0) \ge d_{f_k}(X_0).$$

Now we select a constant value c and the most important subset of faults $\mathit{F} \subset F$ by $\mathit{F} := \{f_i \mid i \le c\}$, the results presented in the next section are provided by c=20. If as usual the number of faults with low detectability is small enough, then a global optimum can be achieved.

c) Starting partitioning:

Firstly we are looking for a starting partitioning $F_a, F_b \subset \mathit{F}$:

```
1)  Set Fa, Fb := Ø .
2)  For i := 1 to c do
    if D(Fa∪{fi},N,X0,0)+D(Fb,N,X0,0) >
         D(Fa,N,X0,0)+D(Fb∪{fi},N,X0,0)
    then    Fa := Fa∪{fi}
    else    Fb := Fb∪{fi}.
```

The so achieved starting partitioning corresponds to an objective value $v := D(F_a,N,X_0,0) + D(F_b,N,X_0,0)$.

d) Constructing a search tree:

Now a search tree T can be constructed, where each node represents two disjoint subsets of F. Node A is a direct successor of node B, if one of the subsets of A is equal to one of B, and if the other subset contains exactly one more fault (see figure 3). Thus a node of depth k represents a partitioning the first contiguous faults in the aforementioned enumeration.

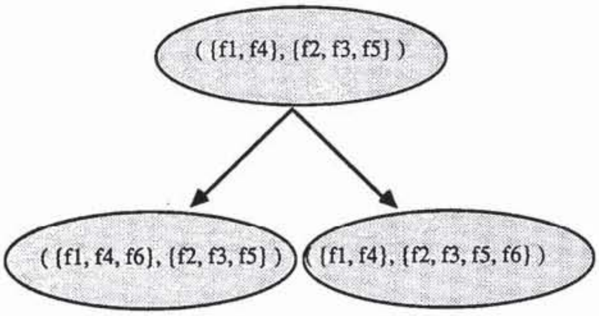

Figure 3: Node at level 5 with its both successors.

Due to the triangle inequation, at a node A at depth m≤c with fault sets F_v, F_w the search can be stopped if

$$v \ge D(F_v,N,X_0,0) + D(F_w,N,X_0,0) + (m-c)d_{fm}(X_0),$$

since no leaf succeeding node A will be better than the starting partitioning. If we reach a leaf thisway, then a better solution F_a, F_b is found and v must be updated. The complexity of this proceeding is distinctly lower than 2^c, since most of the branches are aborted at a very early stage of the search.

e) The complete partitioning:

The resting faults of F (if some exist) are now added to the sets F_a and F_b in the same way as described in c). Finally elements are exchanged between F_a and F_b such that the value of

$$v := D(F_a,N,X_0,0) + D(F_b,N,X_0,0)$$

is maximized.

f) Computing a tupel of optimal input probabilities.

If the gradient for

$$\delta_N^F$$

is already computed, formula (20) is solved by a bisection method. This provides a γ_a with $D(F_a,N,X_0,\gamma_a)=0$ and a γ_b with $D(F_b,N,X_0,\gamma_a)=0$. We set

$$F_1 := F_a, \quad V_1 := X_0 - \gamma_a \mathrm{grad}(\delta_N^{F_1})(X_0), \quad \text{and}$$

$$F_2 := F_b, \quad V_2 := X_0 - \gamma_b \mathrm{grad}(\delta_N^{F2})(X_0).$$

Finally we improve V_1 and V_2 by the procedure OPTIMIZE of section 3. If the gradient is unknown this is done immediately.

5) Multiple optimal distributions

Of course partitioning is not restricted to two sets. But instead of partitioning into m sets at one time, experience has shown better results by a successive procedure:

```
Procedure    Multiple_Optimize
             (F: Faultsets, X: Startvector,
             m: Number of distributions).
    F[1] := F
    X[1] := X
    For i :=1 to m-1 do
         Find fault f with lowest
         detection probability.
         Let j ≤ m-1 be such that f ∈ F[j].
         Partition F[j] into Fa, Fb.
         Optimize (Fa,x[j],xa) and Optimize
         (Fb,x[j],xb) as mentioned in
         sect.4e).
         F[j] := Fa, x[j] := xa, F[i] := Fb,
         x[i] := xb.
```

6) Results

In table 1 optimizing results are shown based on PROTEST. The first example is the ANDOR32-circuit of figure 2. For the wellknown benchmark circuits [Brgl85], k = 1,...,6 optimized input probabilities have been computed. The first column denotes the circuit name, the second one the necessary number of not optimized, equiprobable random patterns, and the following 6 columns contain the necessary number of random patterns for each distribution and its sum. For the set of distributions which results in a minimal size, the overall number of test patterns is printed by bold letters.

For the small circuit C17 the some distributions degenerate to deterministic test patterns. Especially three points are remarkable:

Firstly, all of the benchmark circuits and the counter-example can be tested by only few thousands of random patterns. From a theoretical point of view, all circuits can be made random testable by the presented procedure.

Secondly, the overall number of necessary patterns does not decrease monotonously with the number of distributions. This is a practical consequence of the discussed problem complexity and the applied heuristics.

Thirdly, the results slightly differ from the results reported in [Wu87a,b], since some parameters of the testability measure have been changed in order to speed up optimizing. This has been paid by less precision, and therefore the predictions on fault coverage and test lengths have to be validated by fault simulation.

Circuit	Necessary number of random patterns						
	equiprobable	weights: 1	2	3	4	5	6
AndOr	3.8e10	3.8e10	1.6e3 820 820	1.8e3 200 820 820	2.1e3 200 250 820 820	2.2e3 200 150 250 820 820	2.4e3 200 150 250 200 820 820
c17	85	58	59 27 32	54 27 10 17	55 19 9 17 10	51 8 7 9 17 10	49 8 7 10 9 8 7
c432	1.9e3	1.1e3	1.1e3 4.6e2 6.1e2	1.3e3 4.6e2 4.3e2 4.0e2	1.2e3 4.0e2 4.6e2 1.6e2 1.9e2	1.1e3 1.9e2 1.6e2 4.0e2 2.0e2 1.6e2	1.3e3 1.6e2 1.6e2 1.9e2 2.0e2 2.7e2 2.9e2
c499	1.2e3	1.2e3	2.3e3 1.1e3 1.2e3	3.0e3 1.1e3 1.1e3 8.5e2	3.7e3 1.1e3 1.1e3 8.2e2 6.9e2	4.0e3 6.9e2 8.2e2 1.1e3 5.5e2 8.3e2	4.2e3 6.9e2 1.1e3 5.5e2 8.3e2 7.7e2 2.7e2
c880	2.4e4	3.4e3	9.2e2 1.1e2 8.1e2	7.1e2 1.1e2 1.7e2 5.3e2	1.3e3 1.1e2 1.7e2 4.9e2 5.1e2	1.2e3 1.1e2 1.7e2 5.1e2 4.6e1 4.4e2	1.1e3 1.1e2 1.7e2 5.1e2 4.6e1 4.9e1 2.5e2
c1355	2.1e6	2.1e6	4.0e6 1.9e6 2.1e6	5.7e6 1.7e6 1.9e6 2.1e6	7.2e6 1.5e6 1.7e6 1.9e6 2.1e6	9.3e6 1.5e6 1.7e6 1.9e6 2.1e6 2.1e6	1.1e7 1.5e6 1.7e6 1.9e6 2.1e6 1.7e6 1.7e6
c1908	5.1e4	5.1e4	5.1e4 2.4e2 5.1e4	3.4e4 2.4e2 2.5e4 9.1e3	2.1e4 2.4e2 9.1e3 1.5e3 1.0e4	2.2e4 2.4e2 9.1e3 1.5e3 5.2e3 6.1e3	2.1e4 2.4e2 1.5e3 5.2e3 6.1e3 3.4e3 4.6e3
c2670	8.8e6	7.2e6	4.7e6 1.0e1 4.7e6	2.2e6 1.0e1 2.1e1 2.2e6	1.6e6 1.0e1 2.1e1 8.1e1 1.6e6	8.1e5 1.0e1 2.1e1 8.1e1 2.1e1 8.1e5	8.1e5 1.0e1 2.1e1 2.1e1 8.1e1 1.4e2 8.1e5
c3540	8.0e5	4.6e5	2.9e5 1.7e3 2.9e5	2.1e5 1.7e3 9.8e4 1.1e5	1.7e5 1.7e3 1.1e5 3.3e1 6.1e4	1.1e5 1.7e3 1.1e5 3.3e1 1.4e3 4.1e2	1.8e5 4.1e2 1.4e3 1.7e3 3.3e1 6.8e4 1.1e5
c5315	4.0e4	1.3e4	1.1e4 2.4e2 1.1e4	1.1e4 2.4e2 8.6e3 2.2e3	1.3e4 2.4e2 2.2e3 7.2e3 2.9e3	1.4e4 2.4e2 2.2e3 2.9e3 1.2e3 7.2e3	1.4e4 2.4e2 2.2e3 2.9e3 1.2e3 3.3e2 7.2e3
c6288	6.3e2	2.4e2	4.0e2 1.6e2 2.4e2	4.7e2 1.6e2 2.2e2 8.7e1	5.2e2 8.7e1 1.6e2 4.9e1 2.2e2	5.6e2 4.9e1 8.7e1 1.6e2 5.1e1 2.1e2	6.1e2 4.9e1 5.1e1 8.7e1 1.5e2 6.8e1 2.1e2
c7552	3.1e11	3.0e11	2.2e9 2.6e2 2.2e9	1.1e8 2.6e2 2.5e2 1.1e8	1.1e8 2.5e2 2.6e2 2.5e3 1.1e8	4.5e6 2.5e2 2.6e2 2.5e3 4.5e6 2.0e2	2.8e5 2.0e2 2.6e2 2.5e3 2.5e2 2.8e5 2.6e3

Table 1: Distributions and test sizes

Recently Schulz and Auth succeeded in identifying all redundant faults of the benchmark circuits by a deterministic test pattern generator [ScAu88]. The first column of table 2 contains the number of redundant faults they recognized, in the second column the number of weighted patterns are given. The third column contains the number of not detected faults by simulation, then the fault coverage with respect to all faults, and finally the fault coverage with respect to all detectable faults.

Circuit	Number of redundant faults by [ScAu88]	Number of simulated patterns	Fault coverage (%)	Number of not detected faults	Fault coverage with respect to all detectable faults (%)
c17	0	23	100.00	0	100.00
c432	4	494	99.32	4	100.00
c499	8	1381	98.99	8	100.00
c880	0	578	100.00	0	100.00
c1355	8	5288	99.53	8	100.00
c1908	9	10784	99.60	9	100.00
c2670	117	47110	95.48	117	100.00
c3540	137	11233	96.30	137	100.00
c5315	59	4430	99.07	59	100.00
c6288	34	239	99.59	34	100.00
c7552	131	24037	98.3	158	99.7

Table 2: Fault coverage by simulation of weighted patterns

It should be noted that the simulated amount of patterns has been much smaller than that one required by PROTEST, due to restrictions of the available fault simulator. Presumably this is the reason for the 0.3 percent not detected faults of the circuit c7552, and thus the weighted patterns will be another application field of the recently proposed fast fault simulators for combinational circuits (e.g. [MaRa88], [Waic85]).

Finally table 3 shows for each circuits the optimal number of distributions and the percentage of the size of an optimized random test set in terms of a conventional one.

Circuit	Optimal number of distributions	Size of an optimized test set in percents of a conventional one
AndOr	2	4.2e-8 %
C17	6	58 %
C432	2	58 %
C499	1	100 %
C880	3	3 %
C1355	1	98 %
C1908	6	41 %
C2670	5	9 %
C3540	5	13 %
C5315	5	28 %
C6288	1	38 %
C7552	6	9.0e-5 %

Table 3 Optimal number of distributions and test sizes

7) Applications

The mentioned tools estimating fault detection probabilities are mainly used to predict the necessary test length of a random test, which can be carried out by a built-in self-test structure like a BILBO [KOEN79]. Since a large class of circuits is resistant to such a conventional random test, optimized input probabilities were computed. They can also be implemented as self-test using a so called GURT (Generator of Unequiprobable Random Tests) [Wu87c]. But even this way not all circuits can be dealt with.

The presented method of computing multiple distributions is applicable to ***all*** combinational circuits, but unfortunately there is no obvious way to implement them by a BIST technique. But of course they can be used for a so called LSSD or scan-path random test ([EiLi83], [BaMc84]). Figure 4 shows the basic architecture.

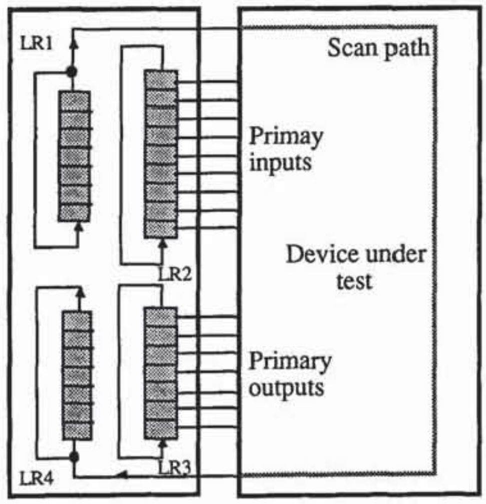

Figure 4: LSSD-based random test

The pattern generator and the signature registers are built on an external chip generating random patterns with multiple distributions sequentially. Such a test chip has been designed and processed as a gate array [Berg85]. Currently a programmable circuit based on standard cells is designed and processed.

This leads to a weighted random pattern test system at low costs, where the same or even a better fault coverage is reachable as it is during a conventional deterministic test. In addition to the low priced test equipment the test application time will also decrease due to the high speed pattern generation.

Conclusions

An efficient method has been presented to compute multiple distributions for random patterns, which can be applied successively. Using multiple distributions, all combinational circuits can be made random testable, and a complete fault coverage is provided by a few thousands of random patterns.

The differently weighted random test sets can be applied to scan path circuits using an external chip, combining the advantages of a low cost test and of high fault coverage.

Furthermore several facts about testing by random patterns have been proven. It has been shown, that the number of random patterns required for a certain fault coverage can be computed without regarding the *pseudo*-random property and with the *independence* assumption for fault detection.

Acknowledgement: I have to thank Mrs. Sybille Hellebrand for many helpful hints, and Mr. Günther Burr and Mr. Klaus Steinheimer for the fast and careful implementations.

Literature:

AaMe87 Aas, E.J.; Mercer, M.R.: Algebraic and Structural Computation of Signal Probability and Fault Detectability in Combinational Circuits in: FTCS-17, Digest of Papers, 1987

AgJa84 Jain, S.K.; Agrawal, V.D.: STAFAN: An Alternative to Fault Simulation in: Proc. 21st. Design Automation Conference, 1984

AkKr84 Krishnamurthy, B.; Akers, S.B.: On The Complexity of Estimating the Size of a Test Set; in: IEEE Trans. on Computers, Vol. C-33, No. 8, 1984

BaMc84 Bardell, P.H.; McAnney, W.H.: Parrallel Pseudo-random Sequences for Built-In Test in: Proc. 1984 International Test Conference

BaSa83 Savir, J.; Bardell, P.H.: On Random Pattern Test Length; in: Proc. 1983 International Test Conference

Berg85 Bergsträsser, T.: Entwurf eines modifizierten Zufallsmustergenerators; Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe, 1985

BRGL85 Brglez, F. et al.: Accelerated ATPG and fault grading via testability analysis; in: Proc. IEEE International Symposium on Circuits and Systems, June 1985, Kyoto

BDS83 Savir, J.; Ditlow, G.; Bardell, P.H.: Random Pattern Testability; in: FTCS-13, Digest of Paper, 1983

ChHu86 Chakravarty, S.; Hunt III, H.B.: On the Computation in: Proc. 21st. Design Automation Conference, 1984

AkKr84 Krishnamurthy, B.; Akers, S.B.: On The Complexity of Estimating the Size of a Test Set; in: IEEE Trans. on Computers, Vol. C-33, No. 8, 1984

BaMc84 Bardell, P.H.; McAnney, W.H.: Parrallel Pseudo-random Sequences for Built-In Test in: Proc. 1984 International Test Conference

BaSa83 Savir, J.; Bardell, P.H.: On Random Pattern Test Length; in: Proc. 1983 International Test Conference

Berg85 Bergsträsser, T.: Entwurf eines modifizierten Zufallsmustergenerators; Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe, 1985

BRGL85 Brglez, F. et al.: Accelerated ATPG and fault grading via testability analysis; in: Proc. IEEE International Symposium on Circuits and Systems, June 1985, Kyoto

BDS83 Savir, J.; Ditlow, G.; Bardell, P.H.: Random Pattern Testability; in: FTCS-13, Digest of Paper, 1983

ChHu86 Chakravarty, S.; Hunt III, H.B.: On the Computation of Detection Probability for Multiple Faults in: Proc. 1986 International Test Conference

EiLi83 Eichelberger, E.B.; Lindbloom, E.: Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test IBM J. Res. Develop., Vol. 27, No. 3, May 1983

GaJo79 Garey, M.R.; Johnson, D.S.: Computers and Intractability - A Guide to NP-Completeness Freeman 1979, San Francisco

IbSa75 Ibarra, O.H.; Sahni, S.K.: Polynomially Complete Fault Detection Problems in: IEEE Trans. on Comp., Vol. C-24, No. 3, 1975

KOEN79 Koenemann, B. et al.: Built-In Logic Block Observation Techniques; in: Proc. Test Conference, Cherry Hill 1979, New Jersey

LBGG86 Lisanke, R. et al.: Testability-Driven Random Pattern Generation; in: Proc. ICCAD, November 1986

MaRa88 Maamari, F.; Rajski J.: A Reconvergent Fanout Analysis for Efficient Exact Fault Simulation of Combinational Circuits; FTCS-18, Tokyo 1988

ScAu88 Schulz, M.H.; Auth, E.: Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques; in: Proc. FTCS-18, Tokyo 1988

Shed77 Shedletsky, J.J.: Random Testing: Practically vs. Verified Effectiveness in: Fault Tolerant Computing Symp. (FTCS-7), 1977

SETH86 Seth, S.C. et al.: An Exact Analysis for Efficient Computation of Random-Pattern Testability in Combinational Circuits; FTCS-16, 1986

WAGN87 Wagner, K.D. et al.: Pseudorandom Testing in: IEEE Trans. Comp., Vol. C-36, No. 3, 1987

Waic85 Waicukauski, J.A. et al.: Fault Simulation for Structured VLSI in: VLSI Systems Design, Dec. 1985

Wu85 H.-J. Wunderlich: PROTEST: A Tool for Probabilistic Testability Analysis; in: Proc. 22nd Design Automation Conference, 1985, Las Vegas

Wu87a H.-J. Wunderlich: Probabilistische Verfahren für den Test hochintegrierter Schaltungen; Informatik-Fachberichte 140, Springer-Verlag 1987

Wu87b H.-J. Wunderlich: On Computing Optimized Input Probabilites for Random Tests, in: Proc. 24th Design Automation Conference, 1987, Miami Beach

Wu87c H.-J. Wunderlich: Self Test Using Unequiprobable Random Patterns in: International Symposium on Fault-Tolerant Computing, FTCS-17, 1987, Pittsburgh

Wu87d H.-J. Wunderlich: The Random Pattern Testability of Programmable Logic Arrays in: Proc. IEEE International Conference on Computer Design, ICCD'87, New York 1987,

WuRo86 H.-J. Wunderlich, W. Rosenstiel: On Fault Modeling for Dynamic MOS Circuits; in: Proc 23rd Design Automation Conference, 1986, Las Vegas