Weighted Random Patterns with Multiple Distributions

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

1) Abstract

It is well known that random test lengths can be reduced by orders of magnitude using biased random patterns. But there are also some circuits resistant to optimizing. In this paper it is shown that this problem can be solved using several distributions instead of a single one. Firstly we compute bounds of the error caused by the assumption that fault detection consists of completely independent events. Secondly we prove a sharp estimation of the error caused by assuming the random property instead of the pseudo-random property of shift register sequences. Finally a heuristic is presented in order to compute an optimal number of random pattern sets, where each set has its specific distribution and its specific size.

2) Test lengths

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. The probability that each single fault of F is detected by N random patterns at least once often is estimated by the formula (1)

$$J_N = \prod_{f\in F}(1-(1-p_f)^N),$$

where p_f is the detection probability of the fault $f \in F$. Of course formula (1) only holds if we assume that the detection of some faults by N patterns forms completely independent events. It neglects such relations as fault dominance and fault equivalence. Therefore some authors try to compute an exact value by means of Markov-theory [BaSa83], but the next theorem shows that formula (1) is indeed a very precise estimation.

__Theorem 1:__ Let G be the probability that each fault of F is detected at least once by N random patterns. Then we have

$$J_N-(1-J_N)\ln(J_N) \le G \le J_N+\ln(J_N).$$

An immediate corollary of this theorem is

__Corollary 1:__ Formula (1) underestimates the confidence of a random test less than $\ln(J_N)$, and for the more dangerous case formula (1) overestimates less than $(1-J_N)\ln(J_N)$.

__Proof of theorem 1:__ Let $\langle f_i\rangle_{1\le i\le n}$ be an enumeration of F where $i < j$ implies $p_{f_i} \le p_{f_j}$. The notation P(A,N) denotes the probability to detect all faults in the set A by N random patterns. Then it is sufficient to show

$$J_N-(1-J_N)\sum_{j=2}^{n}(1-p_{f_j})^N \le P(F,N) \le J_N+\sum_{j=2}^{n}(1-p_{f_j})^N\prod_{k=1}^{j-1}(1-(1-p_{f_k})^N)$$

Set

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

Using the Bayesian formula we have

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

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

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

Thus

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

and since $\delta_1 = 0$

$$\delta_{n+1} \le \sum_{i=2}^{n+1}(1-p_{f_i})^N\prod_{j\le i}(1-(1-p_{f_j})^N)$$

On the other hand since

$$P(\{f_i \mid i\le n\},N \mid \text{no pattern detects } f_{n+1}) \le 1$$

we have

$$\delta_{n+1} \ge \delta_n+(1-p_{f_{n+1}})^N\left(\prod_{i\le n}(1-(1-p_{f_i})^N)-1\right) \ge \delta_n-(1-p_{f_{n+1}})^N(1-J_N) \ge -(1-J_N)\sum_{i=2}^{n+1}(1-p_{f_i})^N$$

This completes the proof since $P(F,N) = J_N+\delta_n$ qed

Theorem 1 and Corollary 1 indicate that the independence assumption is sufficient for statistical investigation. For instance if we have 3 faults with $p_{f1} = 10^{-7}$, $p_{f2} = 5 \cdot 10^{-7}$ and $p_{f3} = 10^{-6}$ then using formula (1) we would need $N = 69\ 10^6$ patterns in order to detect all faults with probability 0 999 The estimation of theorem 1 yields

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

Using theorem 1 it is easily shown that only the few faults with lowest detection probability have impact on the necessary test length This fact has already been observed in [BaSa83]. In [Wu87] it is remarked that all faults can be neglected with detection probability more than 10 times larger than the minimal detection probability

Often it is discussed that the *pseudo*-random property has to be considered, and there are some papers published on this topic [WAGN87]. But for realistic circuits the difference between the test lengths for random tests and for pseudo-random tests is negligible. This fact is an immediate consequence of theorem 2. 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 all patterns.

Theorem 2: 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' random patterns nor by N pseudo-random patterns For $2^{i/2} \ge N$ we have $N = N'$

Proof: Fault detection by random patterns follows the binomial distribution, and we have $\varepsilon = (1 - p)^{N'}$ or $\ln(\varepsilon) = N' \ln(1 - p)$. Estimations with precision of $O(p^2)$ yield $-\ln(\varepsilon) \approx p\,N$ Fault detection by pseudo-random patterns follows hypergeometric distribution, that is

$$\varepsilon = \frac{\binom{2^i - p\,2^i}{N}}{\binom{2^i}{N}} = \frac{(2^i(1-p))!\,(2^i - N)!}{(2^i(1-p) - N)!\,2^i!} = \prod_{k=0}^{N-1} \frac{(2^i(1-p) - k)}{2^i - k} = \prod_{k=0}^{N-1} \left(1 - \frac{2^i}{2^i - k} p\right).$$

This is estimated with precision

$$O\left(\frac{p^2}{1 - 2^{-i/2} + 2^{-i}}\right)$$

by

$$\ln(\varepsilon) \approx \sum_{k=0}^{N-1} \frac{2^i}{2^i - k} p = 2^i p \sum_{k=2^i-N+1}^{2^i} \frac{1}{k}$$

Hence

$$2^i pN \frac{1}{2^i} = pN \le -\ln(\varepsilon) \le 2^i pN \frac{1}{2^i - N + 1}$$

Since

$$2^i N \frac{1}{2^i - N + 1} = N\left(1 + \frac{N-1}{2^i - N + 1}\right) = N + \frac{N^2 - N}{2^i - N + 1} \le N + \frac{2^i - N}{2^i - N + 1} < N + 1$$

we also have $pN \approx -\ln(\varepsilon)$ 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, for all circuits with more than 25 primary inputs, random and pseudo-random pattern sets will exactly have the same size.

Until now we have seen that one of the main concepts of random tests is the computation of fault detection probabilities. Many tools and algorithms were proposed during the past years estimating these probabilities (e.g. [BDS83], [AgJa84], [Wu85], [ChHu86], [AaMe87]). But their precision is limited, since the problem is at least np-hard, which is a simple consequence of the np-completeness of the fault detection problem [IbSa75]. Furthermore 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].

Both facts point out that we cannot expect tools estimating fault detection probabilities with arbitrary high precision neither analytically nor stochastically The intrinsic error also makes useless algorithms computing random test lengths in a very sophisticated way, and the estimations based on theorem 1 and theorem 2 are justified.

Already in [Shed77] it has been observed that the necessary number of random patterns linearly increases with the reciprocal of the minimal fault detection probability Thus in a conventional random test the size of a test set can grow exponentially with the number of inputs. For instance consider an AND32 (fig. 1) where each input is set to "1" with probability x.

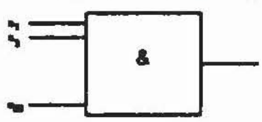

Figure 1: 32 Input AND

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 (1) yields

$$0.999 = (1 - (1 \qquad)^{33}$$

and $N \approx 4.48 \cdot 10^{10}$ But using unequiprobable patterns, i.e. $x \neq 0.5$, test lengths can be reduced drastically ([Wu85], [BGS86]). For example setting

$$x := \sqrt[n]{0.5}$$

we would need approximately $N = 6 \cdot 10^{2}$ patterns.

In [Wu87] an efficient procedure computing optimized input probabilities was presented. But some circuits are resistant to optimizing. For the connection of an AND32 and an OR32 in fig. 2 no solution better than $x = 0.5$ exists.

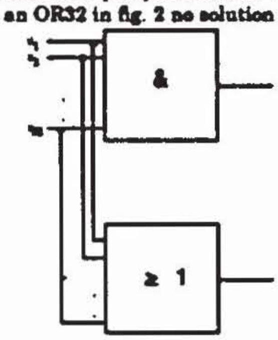

Figure 2: Not random-testable circuit

This problem is solved by applying firstly 600 patterns with $x := 0.5^{1/32}$, and then 600 patterns with $x := 1-0.5^{1/32}$. For the rest of this paper we are dealing with the problem to compute several distributions for random patterns in order to minimize the over-all test length.

3) Optimizing input probabilities

Let $X := \langle x_1,\ldots,x_n \rangle \in [0,1]^n$ be a tupel of real numbers, one number for each primary input. These input probabilities determine the probability for each primary input of being "1", and for each fault they determine its fault detection probability $p_f(X)$ and the probability to detect all faults:

(2)

$$J_N(X) = \prod_{f \in F} (1 - (1 - p_f(X))$$

Now we can try to formulate our problem:

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

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

Immediately it is seen that 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 this problem. Therefore our goal is not an optimal solution, 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. Therefore we formulate our problem as follows:

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

(3)

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

is sufficiently small.

For $k := 1$ this problem has already been solved in [Wu87], and we now list some basic results of this paper For the input probabilities $X := \langle x_1,\ldots,x_n \rangle \in [0,1]^n$ we have for all faults f

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

This is a straightforward consequence of Shannon's formula.

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

By formula (4) and (5) we can compute the 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". By some straightforward approximations formula (3) leads to

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

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

(7)

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

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.

(8)

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

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

(9)

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

Thus the objective function is strictly convex with respect to a single variable, and the explicit formula of (9) 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 :=      2 δN(X)
New :=      δN(X)
While       Old > New + ε    do
            Old  = New
            For i = 1 to n do
                  Search optimal value y for input i.
                  xi := y
                  New  = δN(X)
```

In the next sections we discuss the extension to multiple distributions

4) Partioning of a fault set

Let F be a fault set, and let $X \in [0,1]^I$ be a tupel of input probabilities. 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

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

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 the objective function is maximal. The next theorem will give a helpful hint.

<u>Theorem 3:</u> Let $U \subset R^n$ be convex, $\xi: U \to R$, and let

$$\mathrm{grad}(\xi) := \left(\frac{d\xi}{dx_i}\right)_{1\le i\le n}$$

be the gradient of ξ. For each $x_0 \in U$ the vector $-\mathrm{grad}(\xi)(x_0)$ indicates the direction of strongest decrease. If ξ is linear a local minimum is found on the line $x_0 - \alpha\,\mathrm{grad}(\xi)(x_0)$, $\alpha \ge 0$

<u>Proof:</u> 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

$$\zeta_N^{F^*}: R^+ \cup \{0\} \to R$$

$$\zeta_N^{F^*}(\alpha) := \delta_N^{F^*}(X_0 - \alpha\,\mathrm{grad}(\delta_N^{F^*})(X_0))$$

The formula

(10)

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

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

(11)
$$D(F^*,N,X_0,g) = 0$$

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

(12)
$$D(F_1,N,X_0,0) + D(F_2,N,X_0,0) > 0$$

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.

The gradient

$$\left(\frac{d\delta_N^F(X)}{dx_i}\right)_{i=1}$$

can be computed explicitly using formula (7). If additionally formula (4) is used, it is immediately seen, that we only have to compute $p_f(X)$ and $p_f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$ or $p_f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$ for this purpose.

In order to partition F, for each fault let

$$d_f(X_0) = \sqrt{\sum_{i=1}^{n}\left(\left(\frac{dp_f(X)}{dX_i}\right)(X_0)\right)^2} = \| \mathrm{grad}(p_f)(X_0) \|$$

be the Euclidian norm of the gradient of $p_f(x)$ in x_0, and let $\langle f_i \rangle_{i \le k}$ be an enumeration of F with

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

Now we are looking for a starting partitioning F_a, F_b.

1) Set $F_a, F_b = \emptyset$

2) For i = 1 to k do
   if $D(F_a \cup \{f_i\},N,X_0,0) + D(F_b,N,X_0,0) > D(F_a,N,X_0,0) + D(F_b \cup \{f_i\},N,X_0,0)$
   then $F_a = F_a \cup \{f_i\}$ else $F_b = F_b \cup \{f_i\}$

Starting with this already good partitioning 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. For small fault sets F a search tree T can be constructed computing an optimal partitioning.

After partitioning we have to compute new distributions, one for each new subset of faults. Since the gradient of

$$\delta_N^F(X)$$

is already computed, formula (11) is solved by a bisection method, and subsequently the procedure OPTIMIZE of section 3 is used. 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

```
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 F_a, F_b.
        Optimize (F_a,x[j],x_a) and Optimize (F_b,x[j],x_b) as mentioned in sect. 4c)
        F[j] := F_a, x[j] := x_a, F[i] := F_b, x[i] := x_b
```

6) Applications and results

The mentioned tools estimating fault detection probabilities are mainly used to predict the necessary test length for a random test. It 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) [Wu87a]. But even this way not all circuits can be dealt with.

The presented method of computing multiple distributions is applicable to *all* conventional 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 ([EiLi83a], [BaMc84]), where the patterns are applied to the scan path and to the external inputs of a circuit by an external chip. Currently such a chip is being processed, it is programmable in order to support 4 different distributions.

In table 1 optimizing results are shown based on PROTEST [Wu85]. The results slightly differ from the results reported in [Wu87], since some parameters of the testability measure have been changed in order to speed up optimizing. For the wellknown benchmark circuits [Brgl85], k = 1, 2 and 4 optimized input probabilities have been computed. The first column denotes the circuits name, the second one the necessary number of not optimized, equiprobable random patterns, and the following columns contain the necessary number of random patterns for each distribution and its sum.The first example is the ANDOR32-circuit of fig. 2. It is seen that *all* circuits can be made random testable requiring only few thousands of patterns.

Circuit	Pattern number									
	not optimized	1 distribution			2 distr.					4 distr.
AndOr	$2.5 \cdot 10^{10}$	$2.5 \cdot 10^{10}$	830	820	Σ1640	620	300	250	820	Σ2090
C17	58	41	13	19	Σ32	7*	6*	11	7*	Σ31
C432	2.000	710	390	390	Σ880	150	210	190	160	Σ710
C499	1300	1300	1300	180	Σ1300	1200	180	170	130	Σ1680
C880	28.000	390	160	290	Σ420	180	130	80	90	Σ480
C1355	$6.1 \cdot 10^5$	$6.0 \cdot 10^5$	$6.0 \cdot 10^5$	$5.4 \cdot 10^5$	$\Sigma 11.1 \cdot 10^5$	$6.0 \cdot 10^5$	$5.4 \cdot 10^5$	$5.4 \cdot 10^5$	$5.4 \cdot 10^5$	$\Sigma 22.2 \cdot 10^5$
C2670	$2.5 \cdot 10^6$	$1.4 \cdot 10^6$	$1.4 \cdot 10^6$	$3.6 \cdot 10^5$	$\Sigma 1.8 \cdot 10^6$	$2.4 \cdot 10^5$	$1.3 \cdot 10^3$	$2.0 \cdot 10^3$	$3.0 \cdot 10^4$	$\Sigma 2.7 \cdot 10^5$
C3540	$7.9 \cdot 10^5$	$2.5 \cdot 10^5$	$1.4 \cdot 10^5$	$2.5 \cdot 10^5$	$\Sigma 3.9 \cdot 10^5$	$1.4 \cdot 10^5$	$1.5 \cdot 10^5$	$2.6 \cdot 10^3$	$7.7 \cdot 10^3$	$\Sigma 1.8 \cdot 10^5$
C5315	$6.0 \cdot 10^4$	$3.8 \cdot 10^4$	$3.8 \cdot 10^4$	$3.9 \cdot 10^3$	$\Sigma 4.2 \cdot 10^4$	$3.8 \cdot 10^4$	$3.5 \cdot 10^3$	$7.0 \cdot 10^2$	$2.9 \cdot 10^4$	$\Sigma 7.0 \cdot 10^4$
C7552	$3.4 \cdot 10^{13}$	$4.3 \cdot 10^5$	$6.0 \cdot 10^4$	$3.7 \cdot 10^5$	$\Sigma 4.3 \cdot 10^5$	$6.0 \cdot 10^4$	$8.0 \cdot 10^4$	$3.7 \cdot 10^5$	$4.9 \cdot 10^4$	$\Sigma 5.5 \cdot 10^5$

Table 1: Distributions and test sizes

For the small circuit C17 the marked distributions degenerate to deterministic test patterns. For different circuits there is a different number of distributions in order to minimize the test length. Table 2 shows for each circuit 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	$4 \cdot 10^{-6}$ %
C17	2	55 %
C432	2	33 %
C499	1	100 %
C880	1	1,4 %
C1355	1	98 %
C2670	4	9,5 %
C3540	4	23 %
C5315	1	70 %
C7552	2	$1 \cdot 10^{-6}$

Table 2: Optimal number of distributions and test sizes

Conclusion

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

An efficient method has been presented to compute multiple distributions for random patterns, which have to be applied successively. Using multiple distributions, all circuits can be made random testable. The differently distributed random test sets can be applied to scan path circuits using an external chip, combining the advantages of a low cost test and high fault coverage

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. Comp, Vol. C-33, No.8, 1984

BaMc84 Bardell, P.H.; McAnney, W.H.: Parallel Pseudorandom 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

BRGL85 Brglez, F. et al.. Accelerated ATPG and fault grading via testability analysis; 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

EiLi83a 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

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

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

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; Dissertation, Informatik-Fachberichte 140, Springer-Verlag 1987

Wu87b H.-J. Wunderlich: Self Test Using Unequiprobable Random Patterns; in: International Symposium on Fault-Tolerant Computing, FTCS-17, 1987, Pittsburgh

Wu87c H.-J Wunderlich: On Computing Optimized Input Probabilites for Random Tests; Proc. 24th Design Automation Conference, 1987, Miami Beach