The Design of Random-Testable Sequential Circuits

Hans-Joachim Wunderlich

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

Abstract: In general, sequential circuits are considered not to be random-testable, since a required test sequence may grow exponentially with the number of flipflops, and it is very unlikely that a certain sequence occurs at random. This problem can be solved by combining two tasks:

1) A small part of the flipflops are made directly accessible, for instance by a partial scan path or by a built-in self-test register.
2) Weighted random patterns are applied to the modified sequential circuit.

The paper describes a method to select a minimal set of flip-flops as mentioned in 1). Since this problem turns out to be NP-complete, suboptimal solutions can be derived using some heuristics.

Furthermore, an algorithm is presented to compute the corresponding weights of the patterns, which are time-dependent in some cases. Finally the entire approach is validated with the help of examples. Only 10% - 40% of the flip-flops have to be integrated into a partial scan path or into a BIST-register in order to obtain nearly complete fault coverage by weighted random patterns.

Keywords: Random Test of Sequential Circuits, Built-In Self-Test, Partial Scan Path.

1 Introduction

The random test of integrated circuits has benefits as far as the test application, the test pattern generation and the fault coverage are concerned. One of the most time-consuming tasks in computer-aided testing is the automatic test pattern generation (ATPG). Using random patterns, ATPG becomes superfluous, and a high fault-coverage is ensured, if weighted patterns are applied. There are efficient methods known for computing a single set of weights of a combinational circuit ([Wu85], [LBGG86], [Wu87b]). In some cases, a combinational network may be resistant to a conventional random test, and multiple distributions have to be computed ([Wu88], [WAIC88]). Test patterns corresponding to multiple weights also provide a high coverage of some faults not in the original fault model, for instance, bridging faults and transition faults [WaLi88].

The test application is simplified, if linear feedback shift registers (LFSR) are used to implement a self-test strategy, originally proposed as BILBOs by [KOEN79]. Here, the system registers (Ri) are augmented by some additional circuitry, so that they can generate and evaluate test patterns for the combinational part of the circuit (SNi, figure 1). In [KrAl85], the exhaustive test of pipeline-structures using self-test registers has been proposed. Hence, not all registers are augmented, and the hardware overhead is reduced. This approach will be generalized in the present paper; instead of pipelines we allow more general structures, and the exhaustive test is substituted by a pseudo-random test, with high fault coverage.

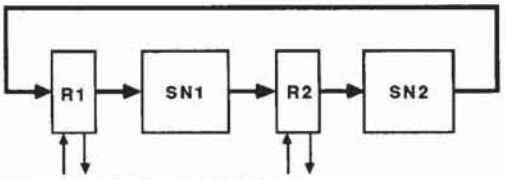

Figure 1: Self-test by LFSRs.

BILBOs generate equiprobable patterns; weighted patterns can be generated during self-test using so-called GURTs (Generator of Unequiprobable Random Tests) [Wu87b]. The hardware overhead of the feedback function of BILBOs and GURTs is avoided, if a scan-path is integrated, and if random pattern generation and evaluation are done off-the-chip (fig. 2).

The migration of the random pattern generation seems to be mandatory, if multiple weights are used. Random patterns corresponding to multiple distributions can be generated on-line by some low cost test-equipment as proposed in [WAIC88], [Strö88]. The implementation of this test strategy requires at least the integration of a complete scan path, which costs additional silicon area, too.

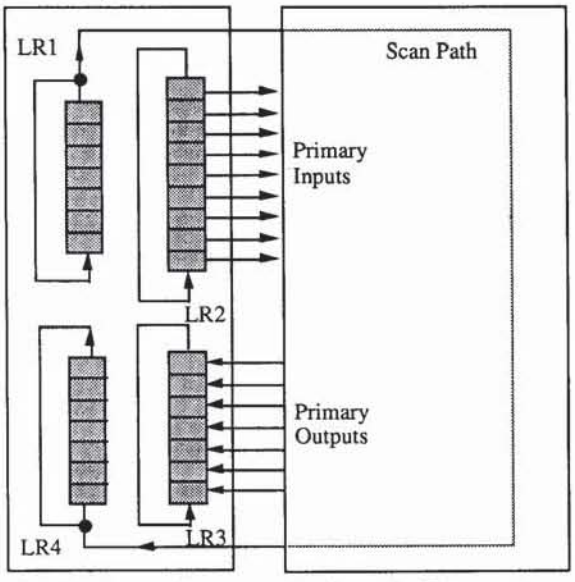

Figure 2: External random test by [BaMc82].

In this paper, we weaken this requirement. We present design algorithms which select a minimal number of flipflops, in order to make a random test feasible for sequential circuits. These flipflops can be integrated either into a partial scan-path or into a GURT or BILBO for a self-test application.

In section 2, we discuss random test lengths for combinational and sequential circuits. A result is that sequential circuits are not randomly testable in general, since the test lengths can grow with a complexity of $O(2^{(2^n)})$, where n is the number of flipflops.

In section 3, we discuss some properties of sequential networks, which ensure bounded test lengths. In section 4, we estimate fault detection probabilities of these modified sequential circuits based on well-known methods for combinational circuits. Moreover, optimal weights of the random patterns are computed. It turns out that the best results are obtained by time-dependent weights at the primary and pseudo-primary inputs of the sequential network.

In section 5, we show that the time-dependence of the weights can be weakened either by a recomputation of the weights which implies longer tests or further design restrictions. Both approaches lead to self-testable sequential circuits. In section 6, we present algorithms which select the scanned flipflops automatically. Finally, we present some results obtained by several example circuits.

2 Random test lengths for combinational and sequential circuits

In this section, we discuss worst-case estimations of test lengths, and show that for some sequential circuits a random test is not feasible, even using multiple weights. Some previous work has been done in estimating the necessary test lengths of combinational circuits based on fault detection probabilities ([ShMc75], [BaSa82], [WaMc86]). Let F be a set of combinational faults, and for each fault $f \in F$ let p_f be its detection probability. Let P(N,F) be the probability to detect all faults of F by N patterns. If fault detection forms completely independent events, then P(N,F) can be estimated by

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

Of course, formula (1) neglects relationships such as fault equivalence and dominance. But in [Wu88] it is shown, that for the actual probability P(N,F) we have

$$(2) \quad J_N - |\ln(J_n)(1-J_N)| \le P(N,F) \le J_N + |\ln(J_n)| .$$

For this reason, formula (1) is precise enough, and it will be used later on. The formula has two important consequences: Only the few faults with lowest detectability determine the necessary test length, and the test length increases linearly as a reciprocal of the minimal fault detection probability. Hence the test lengths may grow exponentially as the number of primary inputs of the combinational network increase.

We consider an AND-gate with n inputs, where each input is set to "1" with probability $x \in [0,1]$. A s0-fault has detection probability x^n, and a s1-fault has $(1-x)x^{n-1}$. Hence formula (1) provides $J_N = (1-(1-x^n)^N)(1-(1-(1-x)x^{n-1})^N)^n$. In order to achieve a test confidence of $J_N = 0.999$, assuming $n = 32$ inputs and equiprobable patterns with $x = 0.5$, approximately $N \approx 4.48 \cdot 10^{10}$ patterns are necessary. A larger number of random patterns are necessary even for the *single* input sequential circuit of figure 3.

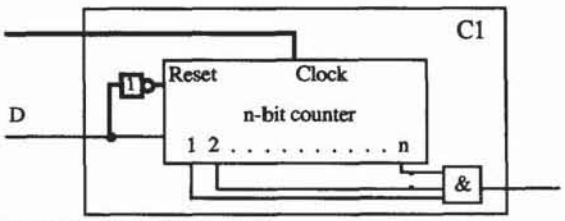

Figure 3: Example circuit C1.

The n-bit counter counts the 1's at the D-input, and it is reset if D = 0. Thus the circuit C1 checks whether there was a "1" at the single data-input D, $(2^n - 1)$ times. The random occurence of such a sequence has a probability of $2^{-(2^n - 1)}$. This 1-input circuit requires a test length similar to a 2^n-input AND. For combinational circuits, the random test length can be reduced by optimal weights [Wu85], [LBGG86], [Wu87]. For instance, setting all input probabilities to $x := \sqrt[32]{0.5}$, we would need only 600 patterns for an AND32. However, this is not a solution for the sequential circuit in figure 4. For this circuit there is no better input probability than D = 0.5.

A similar situation is possible for combinational networks; for instance, if the inputs of an AND32 and an OR32 are connected, then the problem is solved by applying 600 patterns with input probability $x := \sqrt[32]{0.5}$ at first, and later 600 patterns with $x := 1 - \sqrt[32]{0.5}$.

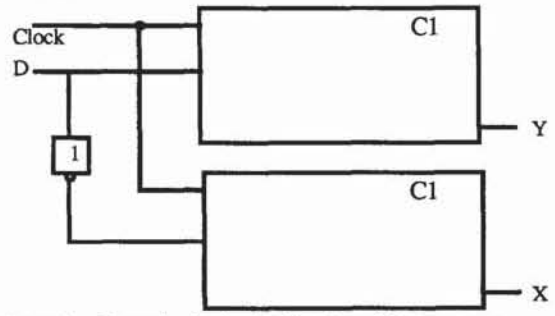

Figure 4: Example circuit checking 1- and 0- sequences.

Methods to compute multiple sets of weights have been presented in ([Wu88a], [WAIC88], and [Wu88b]). But for sequential circuits, the number of weights needed can grow exponentially. An example circuit is given in figure 5.

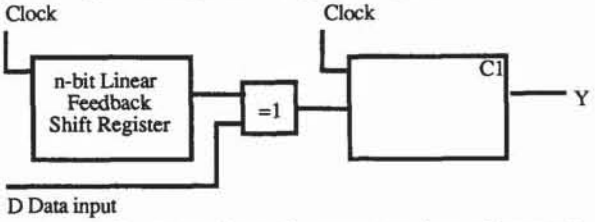

Figure 5: Circuit checking an input sequence by a reference of an LFSR-sequence.

If the LFSR represents a primitive polynomial, then the s0-fault at Y has the detection probability $2^{-(2^n - 1)}$ assuming equiprobable patterns at D. Even a small LFSR with n = 6 requires more than 10^{20} patterns. Since at each time step, the necessary value at

D is determined by the pseudo-random output of the LFSR, $O(2^n - 1)$ different weights are required, in order to reduce the test length significantly.

These simple examples prove, that there is no hope for a weighted random test strategy applicable to all sequential circuits. In the next section we establish some restrictions on the circuit structure ensuring random-pattern testability.

3 Design requirements of random-testable sequential circuits

We assume that the sequential circuits are described at gate level, and that the following restrictions are fulfilled:

- The circuit is purely synchronous.
- Only D-flipflops are used.
- Only one the following conditions holds:
  a) The D-flipflops can be augmented according to the rules of either level-sensitive or edge triggered scan-design (LSSD, ETSD).
  b) The D-flipflops can be augmented to self-test registers, e.g. GURTs or BILBOs.
- For ETSD-circuits, the test signal T blocks the clock of the unscanned flipflops. For LSSD, shift clocks and system clocks must be separated

The main work is to determine a minimal number of flipflops, which have to be augmented according to a) or b), in order to ensure random pattern testability. In order to do this, we have to establish some formal framework:

Definition 1: Let I be the set of (pseudo) primary inputs of a circuit C. $W \subset [0,1]\times I\times\{1,\ldots,n\}$ is a *time-dependent set of weights of length n*, if

a) For all $(i,k) \in I\times\{1,\ldots,n\}$ there is at most one $x\in[0,1]$ with $(x,i,k)\in W$

b) $\forall i\in I\ \exists k\in\{1,\ldots,n\}\ \exists x\in[0,1]\ (x,i,k)\in W$.

$(x,i,k) \in W$ denotes that input i is set to 1 at time-step k, with probability x. There may be some inputs $i\in I$ at some time steps $k\in\{1,\ldots,n\}$, where no weight x is defined. These "don't-cares" are used for compaction later on. We are looking for circuit structures, where a high fault coverage can be obtained using weights of short lengths. If already the deterministic test sequences of a circuit are exponentially long, then a random test will be even longer. Hence, some necessary and sufficient conditions are established to bound the deterministic test lengths, and it is shown, how these conditions also hold for a random test.

We assume that the circuit structure is given by a formal representation, which is transformed into a graph-theoretical form (figure 6).

Definition 2: A circuit graph G := (V,E) is a directed graph with vertices V and edges $E\subset V^2$. $V := V_c\cup V_s\cup I$ is a disjoint union of V_c (vertices corresponding to a combinational element, V_s (vertices corresponding to a sequential element) and inputs I.

The outputs of gates are represented by V_c, the outputs of flipflops are represented by V_s, and I contains both primary and pseudo-primary inputs. The pseudo-primary inputs correspond to the flipflops of the scan-path or the self-test register. We have $(v,w) \in E$, if node v is the input of a component, gate or flipflop with output node w. The primary outputs are a subset $O\subset V$. For the example circuit of figure 7, the circuit graph G := (V,E) consists of nodes $V := \{e_1, e_2, e_3, k_1,\ldots,k_5,a\}$ and the corresponding edges.

For a circuit graph, the *direct predecessors* of $v\in V$ are denoted by $pd(v) := \{w\in V \mid (w,v)\in E\}$, and the *direct successors* by $sd(v) := \{w\in V \mid (v,w)\in E\}$. We assume $I = \{v \in V \mid pd(v) = \emptyset\}$. The *predecessors* of v are $p(v) := \{w\in V \mid$ there is a path from w to v$\}$, and its *successors* are $s(v) := \{w\in V \mid$ there is a path from v to w$\}$. A *path* ω from u to v is a sequence of vertices $k_0, \ldots, k_n$, with $k_0=u$, $k_n=v$ and $(k_{i-1},k_i)\in E$ for $i=1,\ldots,n$, where n is called the *length* $L(\omega)$.

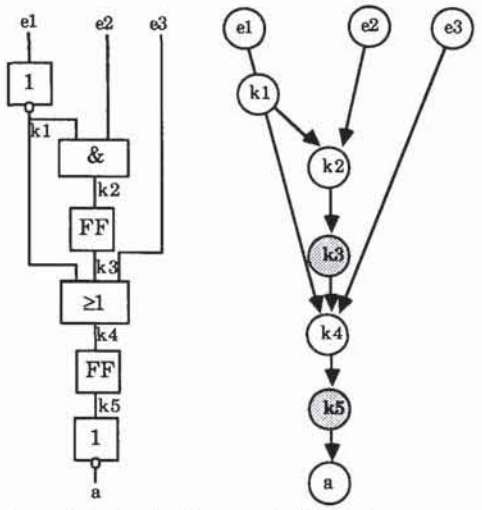

Figure 6: Example circuit and its circuit graph

The topology of the storage elements V_s determines the test length. This topology is described by the so-called S-graph.

Definition 3: Let $G^{Ci} := (V^{Ci},E^{Ci})$ be a circuit graph with $V^{Ci} := V_c^{Ci}\cup V_s^{Ci}\cup I^{Ci}$ and $O^{Ci}\subset V^{Ci}$. Its *S-graph* is defined as $G^S := (V^S,E^S)$, where $V^S := O^{Ci}\cup V_s^{Ci}\cup I^{Ci}$, and $E^S :=$ $\{(v,w)\in V^S\times V^S \mid$ There is a path ω from v to w in G^{Ci}, and $\omega\cap V^S=\{v,w\}\}$.

Figure 7 shows the S-graph corresponding to the circuit graph of figure 6.

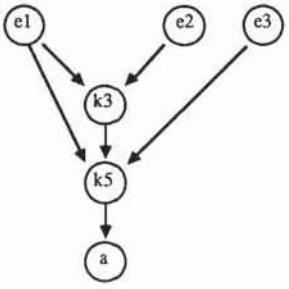

Figure 7: S-graph

The presented approach is valid for a very general fault model. The only restrictions are that no sequential behavior is induced,

for instance, by stuck-open faults and that the topology of the S-graph is not altered, for instance by shorts. The correct circuit and all the faulty circuits are then mapped onto the same S-graph. Since in the faulty case all changes in the functions of the combinational components are admissible, we have to impose some restrictions on the topology of the S-graph, so that the test lengths are linearly bounded. Due to observation 1 below, a necessary condition is that the S-graph contains *no cycles*.

Observation 1: If the S-graph of a sequential circuit contains cycles, the initialization sequence of some states can increase exponentially with the number of flipflops.

A simple example of this observation is a linear feedback shift register (LFSR) of length n, which might have an initialization sequence of length 2^n-1. In [KuWu89] it is also observed that there are S-graphs containing a single cycle, where the length of initialization sequences increases quadratically. On the other hand, we can prove that test sequences are linearly bounded for acyclic S-graphs. A test sequence of a sequential circuit must drive the faulty and the fault-free circuit into states s_f or s, where the responses to the same pattern are different. Hence the maximal test length is given by the maximal necessary state transition sequence. Using Roth's notation of time-frames, copies of the combinational part of the circuit are generated, and the number of time steps corresponds to the length of the test sequence.

We modify this approach, and at each time step, we only copy the small part of the combinational circuit that is actually needed for fault detection. This results in a small combinational representation of the sequential circuit. There is a test sequence for the sequential circuit if and only if, there is a test pattern for its combinational representation. Hence, the computation of fault detection probabilities of sequential circuits is reduced to combinational ones.

In order to describe our solutions exactly, some more graph-theoretical definitions are required. We define the numbers $rf(v) := \max\{\ell(\omega) \mid \omega$ is a path in G with end point v$\}$ and $rb(v) := \max\{\ell(\omega) \mid \omega$ is a path in G with start point v$\}$. We call the maximal path length of an acyclic graph G its *rank* :

$$\text{rank}(G) := \max_{v \in V}\{rf(v)\} = \max_{v \in V}\{rb(v)\}.$$

Definition 4: Let G := (V,E) be an S-graph with sequential nodes V_S, outputs O and inputs I. Its *back-trace function* P is

$$P: \mathcal{P}(V_S \cup I) \rightarrow \mathcal{P}(V_S \cup I);\ P(W) := \bigcup_{w \in W} pd(w).$$

The nodes of a subset $W^t \subset V$ have defined values at time step t, if the nodes $W^{t-1} := P(W^t)$ have defined values at time step t-1, and we can use this notation for state-back-tracing. P^r denotes the application of P r times.

Theorem 1: Let G := (V,E) be an acyclic S-graph with rank(G) = r. Then $P^r(V_S \cup I) \subset I$.

Proof: By the strict decrease of the function rb.

Corollary: Every state is reachable within r steps, if it is reachable at all.

Definition 5: Let $G^S := (V^S, E^S)$ be an acyclic S-graph, with rank r and let $G^{Ci} := (V^{Ci}, E^{Ci})$ be its circuit graph. Set

$W^r := \{v \in V^S \mid sd(v) \cap O \neq \emptyset$ in $G^S\}$,

$V^r := \{v \in V^{Ci} \mid \exists u_0 \in W^r\ \exists u_1 \in O$ (v is member of a path ω from u_0 to u_1 and $\omega \cap V^S = \{u_0, u_1\})\} \cup W^r \cup O$,

and for $0 \le t < r$:

$W^t := P(W^{t+1})$,

$V^t := \{v \in V^{Ci} \mid \exists u_0 \in W^t\ \exists u_1 \in W^{t+1}$ ($v \neq u_1$ is member of a path ω from u_0 to u_1 and $\omega \cap V^S = \{u_0, u_1\}$)$\} \cup W^t$.

The *combinational representation* of G^{Ci} is the graph

$\overline{G} := (\overline{V}, \overline{E})$, where

$$\overline{V} := \bigcup_{0 \le t \le r} V^t \times \{t\}$$

$$\overline{E} := \bigcup_{0 \le t \le r} \{((x,t),(y,t)) \mid (x,y) \in V^t \times V^t \cap E\} \cup \bigcup_{0 \le t < r} \{(x,t),(y,t+1)) \mid x \in V^t \wedge y \in W^{t+1} \wedge (x,y) \in E\},$$

and

$$\overline{V}_c := \bigcup_{0 \le t \le r} \{(x,t) \in \overline{V} \mid x \in V_c^{Ci} \cup V_s^{Ci}\},$$

$$\overline{I} := \{(x,t) \in \overline{V} \mid x \in I\}$$

$$\overline{O} := \{(o,r) \mid o \in O\}.$$

It should be noted, that all flipflops are mapped to combinational buffers. For the example circuit in figure 6 and 7, the time frames are V^2={a, k5}, V^1={k4, k3, k1, e3, e1} and V^0={k2, k1, e1, e2}. The resulting equivalent combinational network is shown in figure 8.

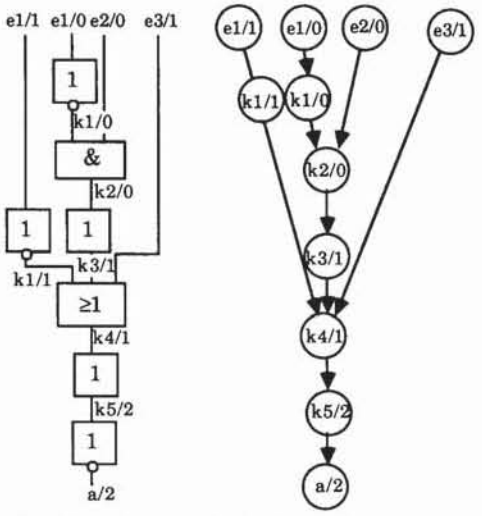

Figure 8: Equivalent combinational network

Each fault at a node v of the sequential circuit corresponds to a multiple fault at the set of nodes $\{(v,t) \mid (v,t) \in \overline{V}\}$ in the combinational representation.

Theorem 2: Let G := (V,E) be an acyclic circuit graph, with rank r, and let $\overline{G} := (\overline{V}, \overline{E})$ be its combinational representation. A pattern sequence $<<b_i^t \in \{0,1\} \mid i \in I> \mid 0 \le t \le r>$ detects a given fault of a node $v \in V$ exactly at time-step r, if and only if the corresponding multiple fault in $\overline{G}$ is detected by the pattern $<b_i^t \mid (i,t) \in \overline{I}>$.

Proof: By construction and theorem 1.

Now we can state the main result of this section:

Corollary: Let G := (V, E) be an acyclic circuit graph, and let $\bar{G}$:= $(\bar{V},\bar{E})$ be its combinational representation. Let $\bar{X}$:= $<(x_{(i,t)},(i,t)) \mid (i,t)\in \bar{I}>$ be a set of weights for $\bar{G}$.

For $v\in V$ let f_v be a fault in G, and let $\bar{f}_v$ the corresponding multiple fault in $\bar{G}$. The probability that a random pattern corresponding to $\bar{X}$ detects $\bar{f}_v$ in $\bar{G}$ is equal to the probability that f_v is detected at time-step r by a random sequence, which is generated by the time-dependent set of weights $X := \{(x,i,t) \mid (x,(i,t))\in \bar{X}\}$.

Since $|\bar{I}| \le r|I|$, the time-dependent set of weights is bounded as required.

4 Test lengths and time-dependent weights.

Computing fault-detection probabilities in combinational circuits is a #-complete problem, and has an exponential worst-case complexity. Hence, algorithms computing them exactly are restricted to a small class of circuits. Estimating procedures may use sampling techniques as STAFAN [AgJa84], may compute bounds like the cutting algorithm [BDS83], or there are analytical techniques, for instance PROTEST [Wu85]. All results reported in this paper are obtained by PROTEST, which is extended in order to deal with combinational representations $\bar{G}$ and multiple faults.

By the last corollary, fault detection at time step r is equivalent to fault detection of the combinational representation. In order to simplify the notation we do not consider fault detection at the first 0,...,r-1 time-steps. Every new pattern $<b_i^t\in \{0,1\} \mid i\in I>$ at time $t\ge r$ provides a new pattern $<b_i^{t-r+k} \mid (i,k)\in \bar{I}>$ for the combinational representation (see fig. 9).

time steps:	0	1	2	3	4	5	6	7
inputs:								
e_1	0	1	0	1	0	1	0	1
e_2	0	0	1	1	0	0	1	1
e_3	0	0	0	0	1	1	1	1

S_1

S_2

S_3

a) Sequences at the sequential network.

(e1,0)	(e2,0)	(e1,1)	(e3,1)
0	0	1	0
1	0	0	0
0	1	1	0
1	1	0	1
0	0	1	1
1	0	0	1
0	1	1	1

b) Corresponding patterns for the combinational representation.

Figure 9: Pattern sequences for the circuit of figure 7 and the corresponding test patterns for the combinational representation.

We assume that the test patterns are completely independent with respect to both the bit position and the time step. In this case the corresponding patterns of the combinational representation are completely independent, too. Let F be the set of faults of the sequential circuit, and let $\bar{F}$ be the corresponding set of (multiple) faults. Finally, let ε be the probability of detecting all faults of $\bar{F}$ by N_c patterns when applied to the combinational representation:

$$(4)\quad \varepsilon \le \prod_{f\in\bar{F}} (1-(1-p_f)^{N_c})$$

Let N_c be the smallest integer fulfilling (4), and let N_s be the necessary number of patterns of the corresponding sequential circuit, then we have

(5) $N_c \le N_s \le N_c+r$, where r is the rank of the S-graph.

Formula (5) holds for equiprobable patterns, and the same result is obtained using time-independent weights $<x_i\in[0,1] \mid i\in I>$ for the sequential circuit. More effort is necessary in analyzing time-dependent weights.

If $W := <(x_{(i,t)},(i,t)) \mid (i,t)\in \bar{I}>$ is a time-dependent set of weights of length n, and $<T_1^1,\ldots,T_n^1>$, $<T_1^2,\ldots,T_n^2>$ are two subsequent pattern sequences corresponding to W, then there are implicitly defined n-1 link weights $L_1,\ldots,L_{n-1}$. They correspond to the n-1 pattern sequences $<T_2^1,\ldots,T_n^1,T_1^2>$, $\ldots,<T_n^1,T_1^2,\ldots,T_{n-1}^2>$.

The link-weights L_j are constructed by augmenting the time-dependent set of weights, such that weights are assigned to all primary inputs at every time-step. Then we do a cyclic shift of the weights by j time-steps, and project the new set of weights to the defined inputs and time steps. This gives us the link-weight L_j.

We proceed in a more formal way, by defining a complementary weight:

Definition 6: Let $W := <(x_{(i,t)},(i,t)) \mid (i,t)\in \bar{I}>$ be a time-dependent set of weights. A set $\bar{W} := <(x_{(i,t)},(i,t)) \mid (i,t)\notin \bar{I}>$ is called a *complementary set.*

Now let $\bar{W}$ be an arbitrary set of weights, complementary to W, and set $U := W\cup\bar{W}$. Define $U^j := \{(x,i,h) \mid (x,i,k)\in U \wedge (h=k+j\le n \vee h=j>n-k)\}$, and set $L_j := \{(x,i,k) \in U^j \mid (i,k)\in \bar{I}\}$.

If we apply N_s+1 pattern sequences corresponding to W, then we also apply N_s pattern sequences corresponding to each link-weight L_j. For each fault $f\in\bar{F}$ the detection probabilities $p_f(W)$ and $p_f(L_j)$, j := 1,...,n-1 depend on the weights of the applied pattern sequences, and formula (4) turns into

$$(6)\quad \varepsilon \le \prod_{f\in\bar{F}} (1-(1-p_f(W))^{N_s+1}\prod_{j=1}^{n-1}(1-p_f(L_j))^{N_s})$$

For small detection probabilities this is estimated by

$$(7)\quad \varepsilon \le \prod_{f\in\bar{F}} (1-(1-(p_f(W)+\sum_{j=1}^{n-1}p_f(L_j)))^{N_s}).$$

Hence we only have to add the corresponding detection probabilities of all time-dependent weights we used, in order to determine the necessary test lengths.

5 Self-testable sequential circuits

Random patterns corresponding to both multiple sets of weights or time-dependent sets of weights can easily be applied using a (partial) scan path and external pattern generator. But in order to implement a self-test, a time-independent, single set of weights seems to be mandatory. Firstly in this section, we discuss time-

independent sets of weights for circuits represented by an arbitrary, acyclic S-graph. Secondly, we discuss design restrictions leading to shorter test lengths.

Definition 7: Let $W := \langle (x_{(i,t)},(i,t)) \mid (i,t) \in \bar{I} \rangle$ be a time-dependent set of weights having a length n. The compaction of W is the time-dependent set of weights $W' \subset [0,1] \times I \times \{1,\ldots,n'\}$, such that:

a) W' has length n';

b) $\forall (x,i,k) \in W\ (x,i,\ k \bmod n') \in W'$;

c) There is no shorter set of weights fulfilling a) and b)

If a pattern sequence is generated corresponding to a time-dependent. compacted set of weights W', then it satisfies the original weights W, too.

For $i \in I$ set $n(i) := |\{k \mid (i,k) \in \bar{I}\}|$, and $h(i) := \sum_{(i,k) \in \bar{I}} x_{(i,k)}$. The average set of weights $W_A := \{(\frac{h(i)}{n(i)},i) \mid i \in I\}$ is time-independent, and patterns corresponding to W_A can be generated by a GURT during a self-test.

W_A is an approximation of the optimal set of weights, and if the result of an average set of weights is not sufficient, further design restrictions are necessary resulting in a pipeline-like structure.

Definition 8: Let G := (V,E) be an acyclic S-graph. G is called *equidistant,* if for all pairs $(u,v) \in V^2$ all paths from u to v have the same length.

The notation of equidistant graphs is a generalization of linear pipelines. A graph is equidistant, if it does not contain any asymmetric reconvergences. Figure 10 gives examples of equidistant S-graphs.

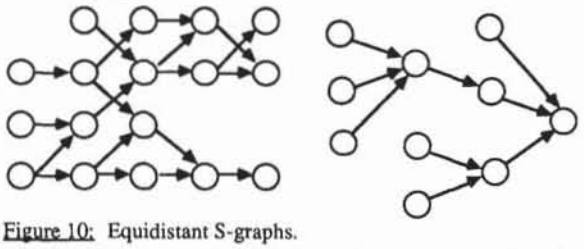

Figure 10: Equidistant S-graphs.

Theorem 3: Let G := (V,E) be an equidistant, acyclic S-graph with a single output o. For each primary input $i \in I$, there is exactly one time step k, such that $(i,k) \in I$ is input of the combinational representation.

Proof: All paths from i to o have the same length.

As a consequence, a time-dependent set of weights W defines a value for a primary input only once. Hence its compaction is $W' \subset [0,1] \times I \times \{1\}$, which actually is a time-independent set.

Definition 9: Let $G^{Ci} := (V^{Ci},E^{Ci})$ be a circuit graph with an equidistant acyclic S-graph. The *combinational reduction* of G^{Ci} is the graph $G^r := (V^r,E^r)$, where $V^r := V'^{Ci}_s \cup V^{Ci} \setminus V^{Ci}_s$, V'^{Ci}_s contains the boolean substitutes of the sequential nodes V^{Ci}_s, and E^r is defined in the obvious way.

In a combinational reduction the flipflops are substitued by lines. In multi-output equidistant S-graphs, for each $o \in O$ we define a subset of inputs $I_o := \{(i,k) \in \bar{I} \mid (i,k) \in p(o)\}$. For each $i \in I$, there is at most one k with $(i,k) \in I_o$. Now we have:

Theorem 4: Let G := (V,E) be an equidistant, acyclic S-graph, let $o \in O$ be an output of the circuit graph, and let $f_v \in F$ be a target fault at node v. A test pattern $T := \langle b_i \mid i \in I \rangle$ detects f_v at output o in the combinational reduction, if and only if the test pattern $T' := \langle t_{(i,k)} \mid t_{(i,k)} = t_i \wedge (i,k) \in I_o \rangle$ detects the fault at output o in the combinational representation.

Proof: Left to the reader.

Now the problem is solved: Fault detection is reduced to fault detection in combinational reductions, and all primary inputs are only needed at the first time-step. Thus we have to compute a single weight at each input, resulting in a time-independent set of weights.

6 Design algorithms

Up to now we have shown that a random test is feasible for a set of weights (at least multiple weights, see [Wu88]), if the sequential circuit is represented by an acyclic or equidistant S-graph. In this section, we discuss how to select the flipflops for self-test registers (GURT or BILBO) or partial scan-paths.

Definition 10: Let G=(V,E) be an S-graph of a sequential circuit. A cut of a node $v \in V$ provides a new graph G'=(V',E'), where

a) $V' = \{p_i\} \cup \{p_o\} \cup V \setminus \{v\}$, with new $p_i \neq p_o \notin V$

b) $E' = \{(p_i,w) \mid w \in sd(v)\} \cup \{(w,p_o) \mid w \in pd(v)\} \cup E \setminus \{(x,y) \mid x=v \vee y=v\}$

If two nodes are cut, then the resulting graph G'' is independent of the order of these cuts. Thus for each $W \subset V$ we can define a graph by $G_W=(V_W,E_W)$. The problem to select a minimal number of scan elements can now be stated as follows:

(FBN): Let G = (V,E) be an S-graph. Find a set $W \subset V$ of minimal cardinality such that $G_W = (V_W,E_W)$ is acyclic.

FBN is known to be NP-complete [Karp72], and heuristics are used in order to obtain good, suboptimal solutions. Let Z_G be the set of all elementary cycles of G. For each cycle $z \in Z_G$, we define $n(z) := \{v \in V \mid v \in z\}$, the set of all nodes of z. Now the scan selection problem is divided into two subproblems:

i) For the S-graph G=(V,E), create the set of all elementary cycles Z_G.

ii) Set $H := \bigcup_{z \in Z_G} n(z)$. Find a set $W \subset H$ of minimal cardinality, such that $\forall z \in Z_G\ W \cap n(z) \neq \emptyset$.

Both subproblems are standard-problems of graph-theory, and there are well-known solutions. The implemented algorithm are based on methods described in [ChKo75], and additional heuristics are used. Alternatively we select a bounded set Z'_G of elementary cycles to solve the hitting problem ii), and select another bounded Z'_G. By a very similar method we can create equidistant S-graphs:

Definition 11: Let G=(V,E) be an acyclic graph. An asymmetric reconvergency between $u,v \in V$ is a set of nodes $R \subset V$, such that

a) There are paths p_1 and p_2 from u to v with $L(p_1) \neq L(p_2)$.

b $p_1 \cap p_2 = \{u,v\}$

c) $R = (p_1 \cup p_2) \setminus \{u,v\}$.

An asymmetric reconvergency R is solved, if one node of R is removed. If Z_G denotes the set of all asymmetric reconvergencies instead of cycles, we have to solve subproblem ii) in the same way as before.

7 Results

We discuss three examples: the operation unit of the signal processor (SP) proposed in [Blan84], a multiplier presented in [Gutb88], and a PROLOG-coprocessor (PP) [Habe87].

Circuit	Inputs	Outputs	Gates	Flipflops
SP	83	55	1675	239
MU	43	26	993	183
PP	36	73	1428	136

Table 1: Circuit characteristics.

The unmodified circuits are hard to test, which is proven with the help of the program LASAR [LASA85]. Fault coverages are listed below obtained after 3600 seconds of computing time.

SP	MU	PP
8.7%	9.8%	11.2%

Table 2: Fault coverage by LASAR after 1h computing time.

In table 3, the percentage of flipflops is given which have to be integrated into a scan-path, in order to generate a complete scan path (CS), equidistant S-graphs (EQ), and acyclic S-graphs (AC).

Circuit	AC	EQ	CS
SP	17.2%	38.5%	100%
MU	39.3%	39.3%	100%
PP	20.6%	44.1%	100%

Table 3: Percentage of elements to be integrated into a self-test register or scan-path.

For the general approach, only 17.2% and 20.6% of the flipflops must be directly accessible. The multiplier (MU) has a structure such that generating an acyclic S-graph automatically provides an equidistant S-graph, too.

By a deterministic test pattern generator it is possible to identify faults undetectable due to redundancies. A sequential redundancy exists, if a fault is not detectable due to unreachable states. For this reason, the number of redundancies will increase from the CS-design over the EQ-design up to the AC-design. Table 4 gives the overall number of redundancies identified by the program SPROUT-9V ([Ku89], [KuWu89]).

	AC		EQ		CS	
SP	4	0.1%	4	0.1%	4	0.1%
MU	10	0.4%	10	0.4%	0	0.0%
PP	692	25.6%	512	18.4%	188	7.0%

Table 4: Total number of redundancies.

Random test patterns are generated and fault coverages are measured only with respect to the remaining faults.

AC-Designs (time-dependent weights): We assume an AC-Design of the example circuits with an integrated partial scan path. Random patterns are generated corresponding to a time-dependent set of weights as described in section 4. Table 5 lists the necessary number of patterns estimated by PROTEST using time-dependent sets of weights and equiprobable patterns sequences.

	SP	MU	PP
weighted	$1.4 \cdot 10^3$	$1.6 \cdot 10^4$	$3.3 \cdot 10^4$
equiprobable	$4.8 \cdot 10^3$	$2.3 \cdot 10^{11}$	$2.2 \cdot 10^6$

Table 5: Estimated random test lenghts (acyclic S-graphs).

The circuit SP needs no time dependent weights, whereas the circuit MU is not testable by a conventional random test. The patterns have been simulated, and the obtained fault coverage is listed in table 6. With the exception of circuit SP, the results are compared with equiprobable patterns.

Using time-dependent weighted patterns, a complete fault coverage is achieved for moderate test-lengths. For the circuits MU and PP, the fault coverage obtained by equiprobable patterns is not sufficient. As already mentioned, the acyclic S-graph of the MU also is equidistant. Thus the used weights are time-independent and can be applied by a GURT. But for the circuit PP we have to determine new time-independent weights.

	SP	MU		PP	
Patterns		weighed	equi	weighted	equi
50	97.4	88.5	92.0	72.9	75.7
100	99.2	92.8	92.8	79.0	80.0
200	99.7	96.1	94.1	84.3	86.2
300	99.8	97.0	94.1	87.4	89.8
400	99.8	97.7	94.4	90.1	91.7
500	99.8	97.9	94.4	91.1	92.4
1000	99.9	98.8	95.2	95.2	95.2
2500	100.0	99.5	95.5	97.7	98.0
3000		99.6	95.5	97.8	98.2
4000		99.9	95.5	98.4	98.8
5000				98.5	98.9
7000				98.9	99.1
9000				99.2	99.3
10000				99.3	99.3
15000				99.7	99.3
20000				99.9	99.3
25000				100.0	99.5
30000					99.5

Table 6: Fault coverage obtained by time-dependent, weighted patterns and by equiprobable patterns.

AC-Designs (average weights): The average weights for the circuit PP have been computed, and the results of fault simulation are listed in table 7.We need a similar test length as before, which could be explained by the fact that no link-weights are necessary for time-independent weights.

Pattern	FC	Pattern	FC	Pattern	FC
50	75.1	1000	97.0	9000	99.9
100	80.7	2500	98.9	10000	99.9
200	87.1	3000	99.2	15000	99.9
300	90.1	4000	99.3	20000	100.0
400	92.2	5000	99.5		
500	93.3	7000	99.7		

Table 7: Fault-coverage obtained by time-independent, weighted patterns.

Up to now, we have derived time-independent weights for all circuits represented by an acyclic S-graph. Only 17.2 % through 39.3 % of the flip-flops must be integrated into a GURT or BILBO, and hence this self-test strategy requires less transistor overhead than the costs of a complete scan-path. Now we try to reduce the test length further.

EQ-Design: The circuits SP and PP have been modified in order to be represented by an equidistant S-graph. The estimated test lengths are listed in table 8.

	SP	PP
weighted	$3.0 \cdot 10^2$	$8.3 \cdot 10^4$
equiprobable	$2.0 \cdot 10^3$	$1.6 \cdot 10^6$

Table 8: Estimated random test lenghts (equidistant S-graphs).

Again, we have generated random patterns and compared the fault coverages obtained by simulation of weighted and equiprobable patterns. Table 9 shows the results.

Pattern	50	100	150	200	250	300
FC	99.2	99.8	99.9	99.9	99.9	100.0

a) Circuit SP

Pattern	FC	Pattern	FC	Pattern	FC
100	91.1	1000	98.9	6000	99.9
200	95.1	2500	99.8	7000	100.0
300	96.1	3000	99.8		
400	97.0	4000	99.9		

b) Circuit PP

Table 9: Fault-coverage obtained by time-independent, weighted patterns.

A complete fault-coverage is obtained by very moderate test lengths.

Conclusions

Algorithms have been proposed, to integrate a small number of flipflops into a partial scan path or a self-test register, but still ensuring random pattern testability. For the modified sequential circuits methods have been discussed in order to compute the necessary test lengths and to determine weights of the random patterns. Here, 3 methods with different trade-offs have been proposed:

1) Time-dependent weights requiring lowest hardware-overhead and short test lengths, but which are not suitable for a self-test strategy.
2) Time-independent weights using acyclic S-graphs, which can be applied during self-test.
3) Time-independent weights using equidistant S-graphs, which require the shortest test lengths and which can be applied during self-test, too. This is paid by slightly more hardware-overhead.

Acknowledgement: The graph-theoretical implementations and investigations on circuit- and S-graphs are due to Arno Kunzmann. I thank Dr. Kunzmann for his help and his comments which where of high value, and I thank Mr. Klaus Steinheimer, who analyzed and simulated all the example circuits.

Literature

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

BaMc82 Bardell, P.H.; McAnney, W.H.: Self-testing of multichip logic modules; in:Proc. 1982 IEEE Test Conf., pp. 200-204

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

Blan84 LeBlanc, J.J.: LOCST: A Built-in Self-Test Technique; IEEE Design & Test, Vol. 1, No. 4,pp.45 - 52

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

ChKo75 Christofides, N.; Korman, S.: A Computational Survey of Methods for the Set Covering Problem; in Management Science, Vol. 21, No. 5, Jan. 1975, pp. 591-599

Gutb88 Gutberlet, P.: Entwurf eines schnellen Matrizen-Multiplizierers; Studienarbeit Fakultät Informatik, Universität Karlsruhe, 1988

Habe87 Haberl, O.: Entwurf und Implementierung eines PROLOG-Preprozessors als Standardzellen-Chip mit dem Entwurfssystem VENUS; Diplomarbeit an der Fakultät Informatik, Universität Karlsruhe, 1987

John75 Johnson, D.B.: Finding all the elementary circuits of a directed graph; SIAM J. Comput., Vol. 4, No. 1, March 1975, pp. 77 - 84

Karp72 Karp, R.M.: Reducibility among combinatorial problems; in R.E. Miller and J.W. Thatcher (eds.), Complexity of Computer, Computations, Plenum Press, New York, pp. 85-103

KrAl85 Krasniewski, A.; Albicki, A.: Automatic design of exhaustively self testing chips with BILBO modules; in: Proc. International Test Conference 1985

Ku89 Kunzmann, A.: Test synchroner Schaltwerke auf des Basis partieller Prüfpfade; Dissertation Fakultät Informatik, Universität Karlsruhe, 1989 (to appear)

KuWu89 Kunzmann, A.; Wunderlich, H.-J.: An Analytical Approach to the Partial Scan Problem; submitted

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

LASA85 ETS LASAR Users Guide, Vers. 4.7, November 1985

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

Roth78 Roth, J.P.: Sequential test generation; Technical Disclosure Bulletin, IBM, Jan. 1978

ShMc75 Shedletsky, J.J.; McCluskey, E.J.: The Error Latency of a Fault in a Combinational Digital Circuit in: Fault Tolerant Computing Symp. (FTCS-5), 1975

Strö88 Ströle, A.: Spezifikation und Implementierung einer Schaltung zur Testmuster erzeugung und Testauswertung als VENUS-Standardzellenchip; Studienarbeit, University of Karlsruhe Dep. of Computer Science

WaLi88 Waicukauski, J.; Lindbloom, E.: Fault Detection Effectiveness of Weighted Random Patterns; in : Proc. 1988 International Test Conference

WaMc87 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

Wu87 H.-J. Wunderlich: Self Test Using Unequiprobable Random Patterns; in: International Symposium on Fault-Tolerant Computing, FTCS-17, 1987, Pittsburgh

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

Wu88 H.-J. Wunderlich: Multiple Distributions for Biased Random Test Patterns; in: Proceedings of the International Test Conference, Washington, 1988

WAIC88 Waicukauski, J.A.; Lindbloom, E.; et al.: WRP: A Method for Generating Weighted Random Patterns; IEEE Design for Testability Workshop, Vail Colorado, 1988