JOURNAL OF ELECTRONIC TESTING: Theory and Applications, 1, 163-174 (1990)

An Analytical Approach to the Partial Scan Problem

ARNO KUNZMANN AND HANS-JOACHIM WUNDERLICH
Institute of Computer Design and Fault Tolerance, University of Karlsruhe, P.O. Box 6980, D-7500 Karlsruhe F.R. Germany

Received July 24, 1989. Revised January 16, 1990.

Abstract. The scan design is the most widely used technique used to ensure the testability of sequential circuits. In this article it is shown that testability is still guaranteed, even if only a small part of the flipflops is integrated into a scan path. An algorithm is presented for selecting a minimal number of flipflops, which must be directly accessible. The direct accessibility ensures that, for each fault, the necessary test sequence is bounded linearly in the circuit size. Since the underlying problem is NP-complete, efficient heuristics are implemented to compute suboptimal solutions. Moreover, a new algorithm is presented to map a sequential circuit into a minimal combinational one, such that test pattern generation for both circuit representations is equivalent and the fast combinational ATPG methods can be applied. For all benchmark circuits investigated, this approach results in a significant reduction of the hardware overhead, and additionally a complete fault coverage is still obtained. Amazingly the overall test application time decreases in comparison with a complete scan path, since the width of the shifted patterns is shorter, and the number of patterns increase only to a small extent.

Key words: design for testability, partial scan path, sequential test generation.

1. Introduction

In 1973, Angell and Williams proposed the scan path in order to facilitate test generation for sequential circuits [3]. In 1977, Eichelberger and Williams established a system of rules called Level-Sensitive Scan-Design (LSSD), resulting in higher flexibility of the design and less hardware overhead [9]. But the costs of a scan path are still up to 20% additional silicon area [4]; and some investigations show significant savings if only a part of the flipflops is integrated into an incomplete or partial scan path.

The partial scan design does not lead to combinational test generation and test application, only the sequential depth of the circuit is reduced. Moreover, the costs of wiring of an incomplete scan path do not directly depend on the number of scan elements. For this reason a partial scan approach is only worthwhile if the number of scan elements can be kept below 50% of all flipflops [34]. Similar results are obtained in [21].

Trischler proposed the selection of scannable elements based on testability measures [35], for instance SCOAP [15]. He uses a general-purpose sequential test pattern generator for the resulting network of reduced sequential depth. But this approach has two main drawbacks:

1. For the modified sequential network, the use of a general ATPG does not guarantee a complete fault coverage.
2. In general, the heuristics for selecting the scannable flipflops do not result in a minimal number.

The first drawback is avoided by the approach presented in [1 and 2]. Here the selection of the scan elements and the generation of the test patterns are integrated into one algorithm. The PODEM algorithm [14] tries to generate test patterns for the combinational part of the circuit; flipflops are selected that should be accessible in order to control the inputs of the combinational part. But the number of selected flipflops will become rather large. Both drawbacks can be avoided by an approach based on the state transition graph as presented in [27], where circuit design and test generation are integrated. This approach is especially applicable for logic synthesis. But, in general, the circuit structure is given and not the state transition graph. Generating this graph for arbitrary networks can become impracticable due to time and storage limits.

In the presented work, both disadvantages are avoided without any serious time and storage restrictions. We establish some necessary restrictions on the circuit structure to ensure that the test length for each fault is linearly bounded by the circuit size. We present a new algorithm selecting a minimal set of scan elements in order to satisfy these restrictions. Since the selection problem is NP-complete, the exact solution optionally is weakened by some heuristics. Based on the selection of the scan elements, a new ATPG-algorithm is developed, which is specially suited for these modified circuits and succeeds in complete coverage of the irredundant faults.

This article is an extension of the work reported in [23]; similar ideas have been presented in [6] and [13].

In section 2 we describe the necessary design restrictions for our approach, introduce the used graph-theoretical circuit representations, and discuss the fault model. After these preliminaries, section 3 points out the relation between circuit structure and test length. The necessary restrictions are established; these must be satisfied by an incomplete scan path in order to bound the test length and the test effort.

Section 4 presents an algorithm for selecting the scannable elements, and section 5 explains the corresponding ATPG-algorithm for the modified network. After a discussion of the necessary and sufficient test lengths and also of the test application time, the results of several benchmark circuits are presented in section 7.

2. Circuit Representation and Fault Modeling

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

- The circuit is purely synchronous.
- The system operation is controlled by a one-phase clock.
- Only D-flipflops are used.
- The D-flipflops can be completed according to the rules of either level-sensitive or edge-triggered scan design (LSSD, ETSD).
- Shifting is controlled by an additional clock, or the system clock for the non-scannable flipflops can be blocked.

The extensions to multiple clocks are omitted in order to simplify the notations. More complex storage elements (e.g., *T*-, RS-, and JK-flipflops) can be used, if these components are modeled by D-flipflops and some combinational logic.

We model sequential circuits by a directed graph, called data-flow graph, where the vertexes or nodes correspond to primary inputs, primary outputs, and to the outputs of the gates and flipflops. There is a directed edge between node v and w, if v is input of a component with output w. For the example circuit of figure 1 the corresponding data-flow graph is given in figure 2. Here the data-flow graph $G = (V, E)$ consists of the nodes $V := \{E1, \ldots, E5, K1, \ldots, K8, K11, K12, P01, P02, P03\}$ and the corresponding edges.

In the next section, it is shown that the topology of the storage elements in particular determines the test length. This topology is described by the so-called S(torage element)-graph. Figure 3 shows the S-graph corresponding to the data-flow graph of figure 2.

The nodes V of the S-graph $G_S := (V, E)$ are the terminals and the outputs of flipflops. There is an edge $(v, w) \in E$ in the S-graph, if there is a path from node v to node w in the data-flow graph, which does not contain any storage element.

The presented approach is valid for a very general fault model, including the classical stuck-at fault model, combinational faults, and most of the bridging faults.

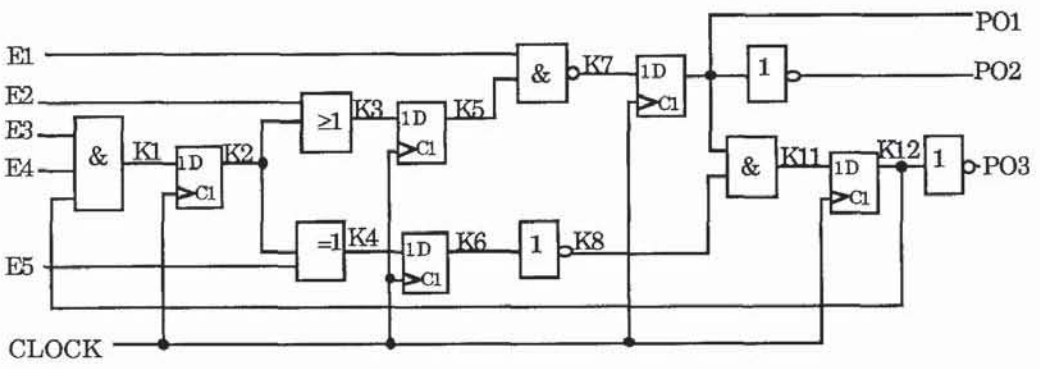

Fig. 1. Example circuit.

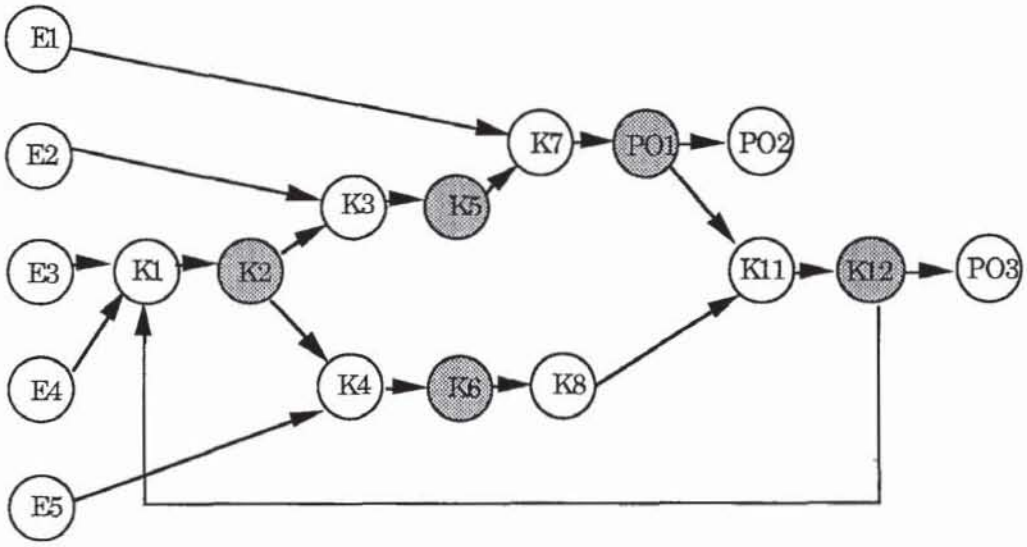

Fig. 2. Data-flow graph.

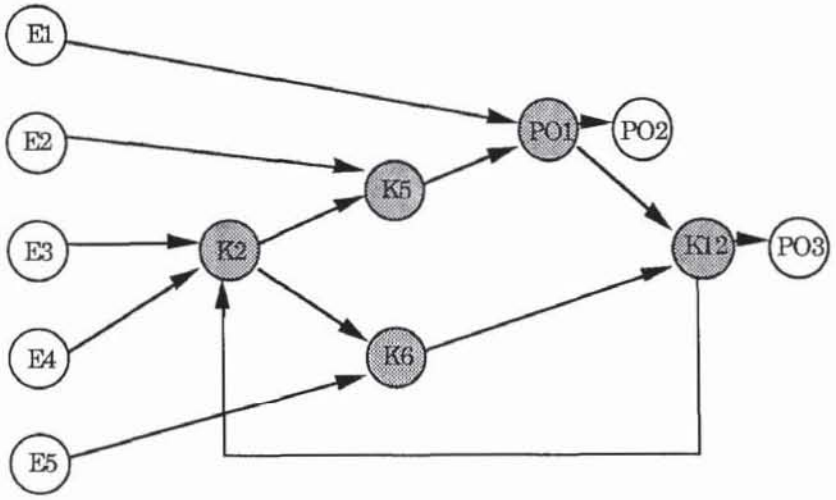

Fig. 3. S-graph.

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 by shorts.

3. The S-Graph and Test Lengths

By the definition of our fault model, the correct circuit and all faulty circuits are mapped onto the same S-graph. Since all faulty changes of the functions of the combinational components are admissible, we have to impose some restrictions on the topology of the S-graph, in order to ensure that the test lengths are linearly bounded. Due to observation 1 and observation 2 below, a necessary condition is that the S-graph contains no cycles (i.e., feedback loops within the gate-level netlist).

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 $2^n - 1$. But even a single cycle can lead to a drastic increase in test size:

Observation 2. There are S-graphs containing a single cycle, where the initialization sequence of some states increases quadratically with the number of flipflops.

Such a circuit is shown in figure 4.

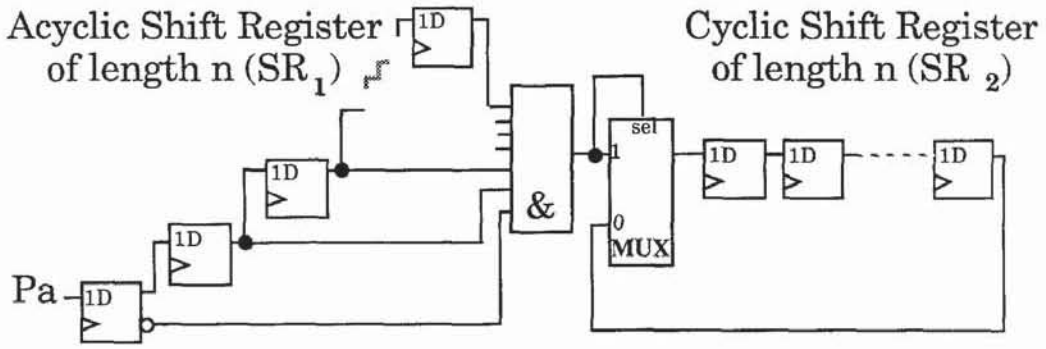

Fig. 4. Sequential circuit with a single cycle.

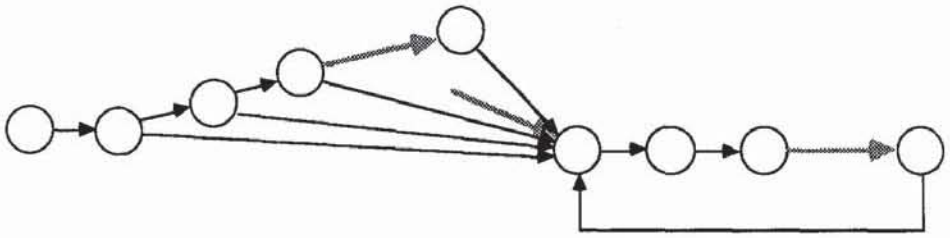

Fig. 5. S-graph of the circuit of Fig. 4.

In order to fill the register SR_2 completely by "1," $O(n^2)$ clocks are required. As illustrated by figure 5, the S-graph contains exactly one cycle only. The actual length of the transition sequence is determined by the data-flow graph, and there are circuits with short test sequences, though the S-graph contains some cycles. But since the data-flow graph is affected by each fault, we have to assume the worst-case, and we demand that the S-graph of a sequential circuit is cycle-free. This condition requires that almost all the flipflops of counters or control parts must be integrated into a partial scan path. This drawback is not too severe, if these flipflops are only a small subset of a total set of flipflops, as we will demonstrate by the analyzed benchmark circuits. On the other hand, it is well known, for example, that large counters are hard to test and they should be directly accessible anyway.

Adding a storage element to an incomplete scan path corresponds to the removal of a node from the S-graph as described in definition 4.1 below. In the next section, we discuss how to determine and to remove a minimal set of nodes in order to get an S-graph without cycles.

In section 5, it is shown that these conditions are not only necessary, but also sufficient, to test a sequential circuit with test pattern sequences of linear bounded length.

4. The Selection of Scan Elements

In order to describe the algorithms selecting scan elements, some graph theoretic notations are necessary. Let $G := (V, E)$ be a finite, directed graph with nodes V and edges $E \subset V^2$. We use the following notations.

A sequence $e_1, \ldots, e_q$ of nodes (vertexes) is a *path*, if $(e_i, e_{i+1}) \in E$ for $i = 1, \ldots q - 1$. The *length* $\ell(\omega)$ of a path ω is the number of directed edges. An *elementary path* is a path where each node occurs only once; and a *cycle* is path where the first node and the last node are identical. An *elementary cycle* is a cycle, where no node occurs twice, with the exception of the first and last node.

Let $G = (V, E)$ be a graph, and $v \in V$ a node. $p(v, G) \subset V$ is the set of all *predecessors* of v in G, $s(v, G) \subset V$ is the set of all *successors*, $pd(v, G)$ is the set of all *direct predecessors*, and $sd(v, G)$ is the set of all *direct successors* of v in G. A node v of the graph G is *initial (final)*, if $p(v, G) = \phi$ ($s(v, G) = \phi$). By this definition, all initial nodes correspond to primary inputs.

Definition 4.1. Let $G = (V, E)$ be an S-graph of a sequential circuit. A *cut* of a node $v \in V$ is a map of G into a new graph $G' = (V', E')$ where

$V' = \{p_i\} \cup \{p_o\} \cup V \setminus \{v\}$, with new $p_i \neq p_o \notin V$
$E' = \{(p_i, w) \mid w \in sd(v, G)\} \cup$
$\{(w, p_o) \mid w \in pd(v, G)\} \cup$
$E \setminus \{(x, y) \mid x = v \vee y = v\}$

If two nodes are cut, then the resulting graph G' is independent of the order in which the cuts are performed. Thus for each $W \subset V$, we can cut all nodes of W, and we denote the resulting graph by $G_W = (V_W, E_W)$.

The problem of selecting a minimal number of scan elements can now be stated as follows:

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

(AC) is an NP-complete problem, sometimes called Feedback Node Problem [20]. For this reason, besides the exact algorithms some heuristics are necessary in order to obtain at least 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 we can divide the scan selection problem into two subproblems (see figure 6):

i) For the S-graph $G = (V, E)$, determine 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. For example, algorithms to solve subproblem (i) are presented in [8, 32, 33, 36]. In the presented approach, an algorithm has been implemented based on [19]. In the worst case, the cardinality $|Z_G|$ may increase exponentially in the size of V. Also subproblem (ii) has a very high complexity, since it is a formulation of the Hitting Set Problem, which is also known to be NP-complete [12, 20].

The implemented algorithms are based on methods described in [7], but in combination with subproblem (ii) additional heuristics are used. These heuristics are divide-and-conquer methods, where the problem size is bounded by a constant $C \in IN$.

This procedure can also be used for probabilistic optimization, since the elementary cycles are selected randomly. Even a single pass provides good results; and for the results reported in section 7, this procedure has been called only once.

In the next section we show that for circuits represented by an acyclic S-graph test patterns can be generated very efficiently.

5. Test Pattern Generation

For general synchronous circuits, Roth introduced the notation of time-frames [29]. For each time step, a copy of the combinational part of the circuit is generated, and the number of time steps corresponds to the length of the test sequence (figure 7).

It is known that for circuits described by an acyclic S-graph the necessary number of time-frames is bounded by the number of storage elements [10]. In this section it is shown that only a small part of the combinational circuit must be copied at each time step. This results in a rather small combinational representation of the sequential circuit. It should be noted, that only a linear number of time-frames is needed in order to identify all redundancies. This includes combinational redundancies and redundancies due to unreachable states. Finally, the test pattern generation algorithm for this

```
PROCEDURE scan_selection;
    SET W:=∅, G_W=(V,E);
    REPEAT
        select a set of elementary cycles Z of G_W at random with |Z_G| ≤ C;
        SET H := ⋃_{z∈Z_G} n(z);
        Solve the hitting problem for H, i.e. find minimal W*⊂H such that
        ∀z∈Z: W*∩n(z)≠∅.
        SET W:=W∪W*;
    UNTIL G_W is acyclic
END scan_selection;
```

Fig. 6. Selection of scan-elements.

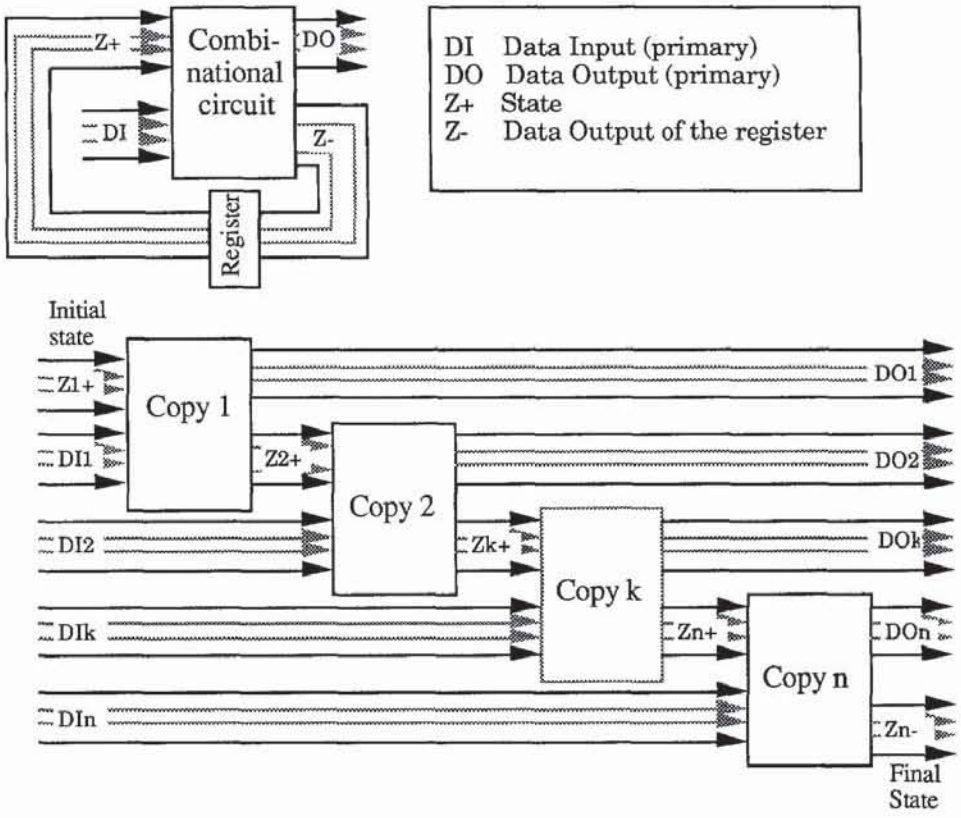

Fig. 7. Time frames.

combinational representation is sketched, and the transformation of the combinational test patterns into sequences is discussed.

5.1. Test Lengths for Acylic S-Graphs

In order to describe our solutions exactly, some more graph-theoretical definitions are required:

Definition 5.1. Let $G = (V, E)$ be an acyclic graph, let $v \in V$ be a node. The value $rf(v) := \max \{\ell(\omega) \mid \omega$ is a path in G with final point $v\}$ is called *forward-rank*, and $rb(v) := \max \{\ell(\omega) \mid \omega$ is a path in G with initial point $v\}$ is called *backward-rank*.

Definition 5.2. Let $G = (V, E)$ be an acyclic graph. The *rank* of G is defined by

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

Let $G = (V, E)$ be an acyclic S-graph with rank$(G) = r$. A time-frame at time t is a subset of nodes $V^t \subset V$. V^r contains all the primary outputs and all the nodes that are connected with a primary output via a path of combinational elements. The time-frames are ordered by $V^{t-1} = \{v \mid v \in pd(w), w \in V^t\}$.

Obviously the nodes of V^t have defined values, if the nodes of V^{t-1} have defined values. Hence we can use this notations for state-back-tracing, and if we find a time-frame V^s, $s < t$, containing only initial nodes, then an upper bound $(t - s)$ of a test sequence is derived.

A test sequence of a sequential circuit must drive the faulty and the fault-free circuit into a state s or s_f, where the responses to the same pattern are different. Hence the maximal test length is given by the maximal required state transition sequence. In a formal way, we can state the following theorem, which was also observed in [10, 18]:

Theorem 5.3. Let $G = (V, E)$ be an acyclic S-graph with rank$(G) = r$. For each flipflop v, there is an initializing sequence of at most length r, if there is such a sequence at all.

Proof. Let $rb(v) = k \leq r$, and start with $V^r = \{v\}$. The theorem is proved if $V^0 = \emptyset$ or V^0 contains initial nodes only. Set

$$m(i) := \max_{u \in V^i} \{rf(u)\},\ i = 0, \ldots, r.$$

By definition 5.1 we have $m(i) = m(i-1) + 1$, if V^i contains storage nodes. Since $m(r) \leq r$ we have $m(0) = 0$. Q.E.D.

In order to get a test sequence, we have not only to set a single flipflop, but we have to assign an entire state. This is possible by the following corollary:

Corollary 5.4. Let $G = (V, E)$ be an acyclic S-graph with rank$(G) = r$. Each admissible state can be reached within r steps.

Proof. Let $S_1 \ldots S_k$ be the flipflops of the circuit, let $f(S_1, \ldots, S_k)$ be a boolean function, which is true, if and only if the desired state is reached. Let S_{k+1} be a new flipflop, and add circuitry such that $S_{k+1} = f(S_1, \ldots, S_k)$. Then the new S-graph has rank $r + 1$, and S_{k+1} can be set within $r + 1$ steps by theorem 5.3. Thus the desired state can be reached within r steps. Q.E.D.

Up to now we have proved that the test lengths are linearly bounded. Now we will describe how to derive the test sequences.

5.2. *Equivalent Combinational Circuits*

For a sequential circuit C with an acyclic S-graph $G := (V, E)$, we are generating an equivalent combinational circuit $\tilde{C}$ of size $O(r(G) \cdot |C|)$, such that a test pattern of $\tilde{C}$ corresponds to a test sequence of C. Moreover, $\tilde{C}$ should be minimized. For this reason we extend our notion of time-frames to data-flow graphs.

Let $G_S := (V_S, E_S)$ be an acyclic S-graph, and $G_D := (V_D, E_D)$ be its data flow graph.

Let V^i be a time-frame of the S-graph. In the data-flow graph, $\tilde{V}^i$ is defined by

$$\tilde{V}^i := V^i \cup \{v \in s(w, G_D) \mid w \in V^i \text{ and there is a combinational path from } w \text{ to } v\}.$$

If one node occurs in several time-frames, it must be copied sufficiently often. Moreover the flipflops are modeled as pseudo-boolean functions in the well-known way.

The equivalent combinational circuit $\tilde{C}$ is represented by the graph $G_C := (V_C, E_C)$ where

$$V_C := \bigcup_{0 \leq i \leq r} \{(v, i) \mid v \in \tilde{V}^i\}$$

and

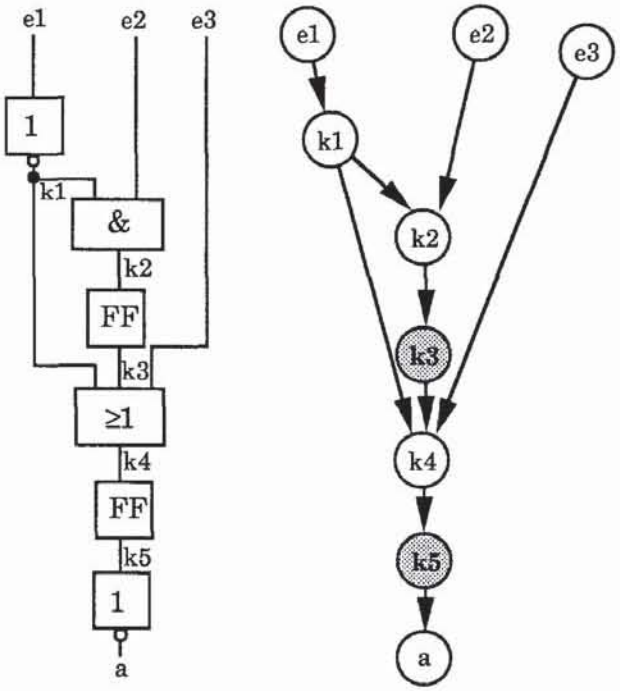

Fig. 8. Sequential circuit and data-flow graph.

$$E_C := \{((v, i), (w, j)) \mid (v, w) \in E_D \wedge (i = j - 1 \wedge w \text{ is a flipflop})\}.$$

Instead of the formal description of this straightforward method, an example is given. Figure 8 shows a sequential circuit and its data flow graph. The marked nodes of the data-flow graph correspond to flipflops. We have the following time-frames:

$$V^2 = \{a, k5\},$$
$$V^1 = \{k4, k3, k1, e3, e1\} \text{ and}$$
$$V^0 = \{k2, k1, e1, e2\}.$$

The resulting equivalent combinational network is shown in figure 9. Test patterns, generated for the combinational representation with 4 inputs, have to be transformed to pattern sequences of length 2 as represented in table 1.

5.3. *Test Pattern Generation*

A target fault may affect a gate, which is represented in multiple time-frames. In this case the fault is copied, and test generation must be done for a combinational network with multiple faults.

For this purpose the algorithm SPROUT-9V (Signal **Pro**bability **U**sing **T**est Pattern Generator, **9**-valued logic) has been implemented. The tool is described in

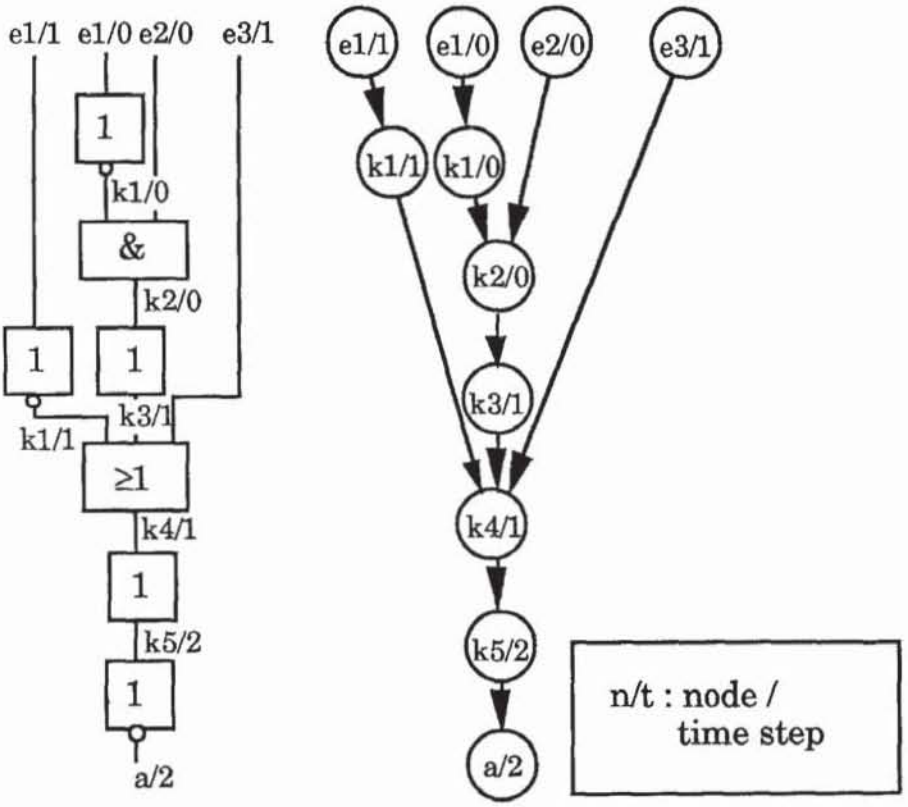

Fig. 9. Equivalent combinational network.

Table 1. Test pattern transformation.

Test pattern					Test pattern sequences		
*e*1/1	*e*1/0	*e*2/0	*e*3/1		*e*1	*e*2	e3
1	0	1	1	⇒	0	1	—
					1	—	1
0	1	0	0	⇒	1	0	—
					0	—	0

detail in [22], and we are only sketching its main features. It is based on the nine-valued algebra proposed in [28], in order to guarantee the generation of test patterns for any detectable fault, if the computing time is not limited. The algorithm is an enumeration algorithm controlled by estimations of signal probabilities derived by PROTEST [37, 38]. Additional heuristics accelerate the TPG in a similar way as used in FAN [11] or SOCRATES [30, 31].

6. Test Application Time

The test application time is determined by both the length of the scan path and the number of pattern sequences. The shifting time is shortened by using a partial scan path, and it turns out that the number of patterns does not increase as fast as the number of scan elements decreases. Thus the overall test application time is reduced.

In many cases it is possible to develop an overlay scheme for the test sequences. That is, new patterns can be applied at a time when parts of the circuit are still dealing with former sequences. Most widely this pattern compaction can be used for pipeline structures. Here, after a certain start-up time, an overlay technique can be used and at each time step a new pattern can be applied. Figure 10 shows the S-graph for pipeline circuits.

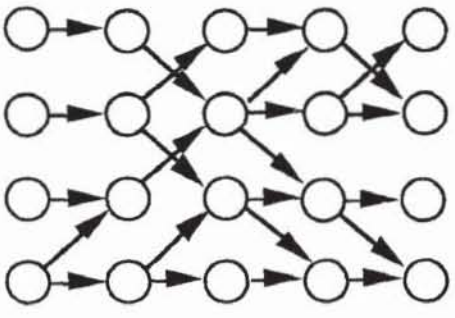

Fig. 10. S-graph for a pipeline structure.

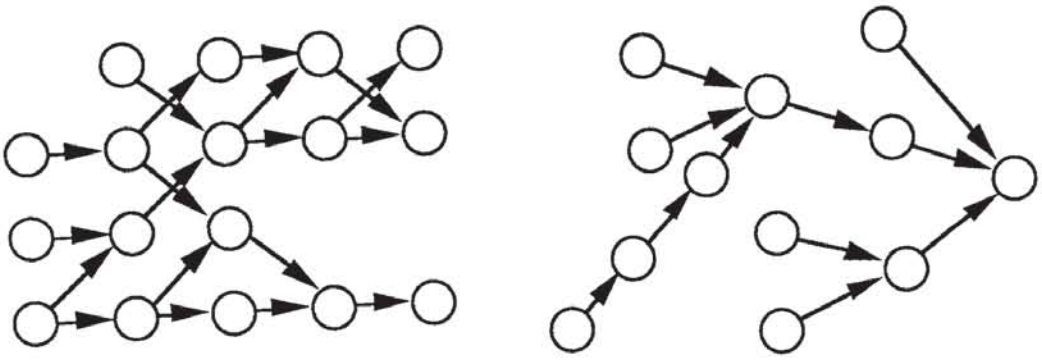

Fig. 11. Equidistant graphs.

A complete overlay of test patterns is also obtainable for more general structures than pipeline circuits. An acyclic S-graph $G = (V, E)$ is called equidistant, if for two arbitrary nodes $u, v \in V$, all paths with initial point u and final point v have the same length. Figure 11 shows two examples.

Definition 6.1. 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

i) there are paths p_1, and p_2 from u to v with $\ell(p_1) \neq \ell(p_2)$.
ii) $p_1 \cap p_2 = \{u, v\}$.
iii) $R = (p_1 \cup p_2) \setminus \{u, v\}$.

An acyclic graph is equidistant if there are no asymmetric reconvergencies. It can be generated by adding a few more elements to the partial scan path.

An asymmetric reconvergency R is solved if at least one node of R is removed by adding the corresponding storage element to the scan path. Searching a minimal set solving all asymmetric reconvergencies is an NP-complete problem (see also [24]). Thus we are using heuristics, and we have to solve the same subproblems as presented in section 3:

i) Create all asymmetric reconvergencies R_G.
ii) Set $K := \bigcup_{R \in R_G} R$.

Find a set $W \subset K$ of minimal cardinality, such that $\forall R \in R_G: W \cap R \neq \emptyset$.

Of course, the shorter test lengths for equidistant S-graphs are at the expense of somewhat longer scan paths.

7. Results and Applications

The presented algorithms are implemented and integrated into a tool system called INSPIRATION (**In**complete **S**can **P**ath **I**nte**g**ra**tion**). Figure 12 summarizes the architecture of this system.

The approach can also be extended to a test by weighted random patterns (WRP). For a design with a complete scan path, the application of weighted random patterns has been discussed in [25]. A partial scan path requires the computation of time-dependent weights as discussed in [39].

Several sequential circuits have been analyzed by INSPIRATION as described in [24]. Here, we discuss three examples. The first one is the operation unit of the signal processor (SP) proposed in [5]. The second example is a multiplier (MU) presented in [16], and we discuss a processor to accelerate PROLOG-programs (PP) [17]. Table 2 shows the relevant data of the example circuits including the transistor count.

Table 2. Circuit characteristics.

	Inputs	Outputs	Gates	Flipflops	Transistors
SP	83	55	1,675	239	21,776
MU	43	26	993	183	14,652
PP	36	73	1,428	136	17,242

Both discussed test strategies have been investigated: If the main objective is to reduce the hardware overhead, one has to generate acyclic S-graphs, and if the objective is to reduce test lengths, one has to generate acyclic equidistant S-graphs. Table 3 gives the percentage of flipflops which has 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).

Table 3. Percentage of scan path elements.

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

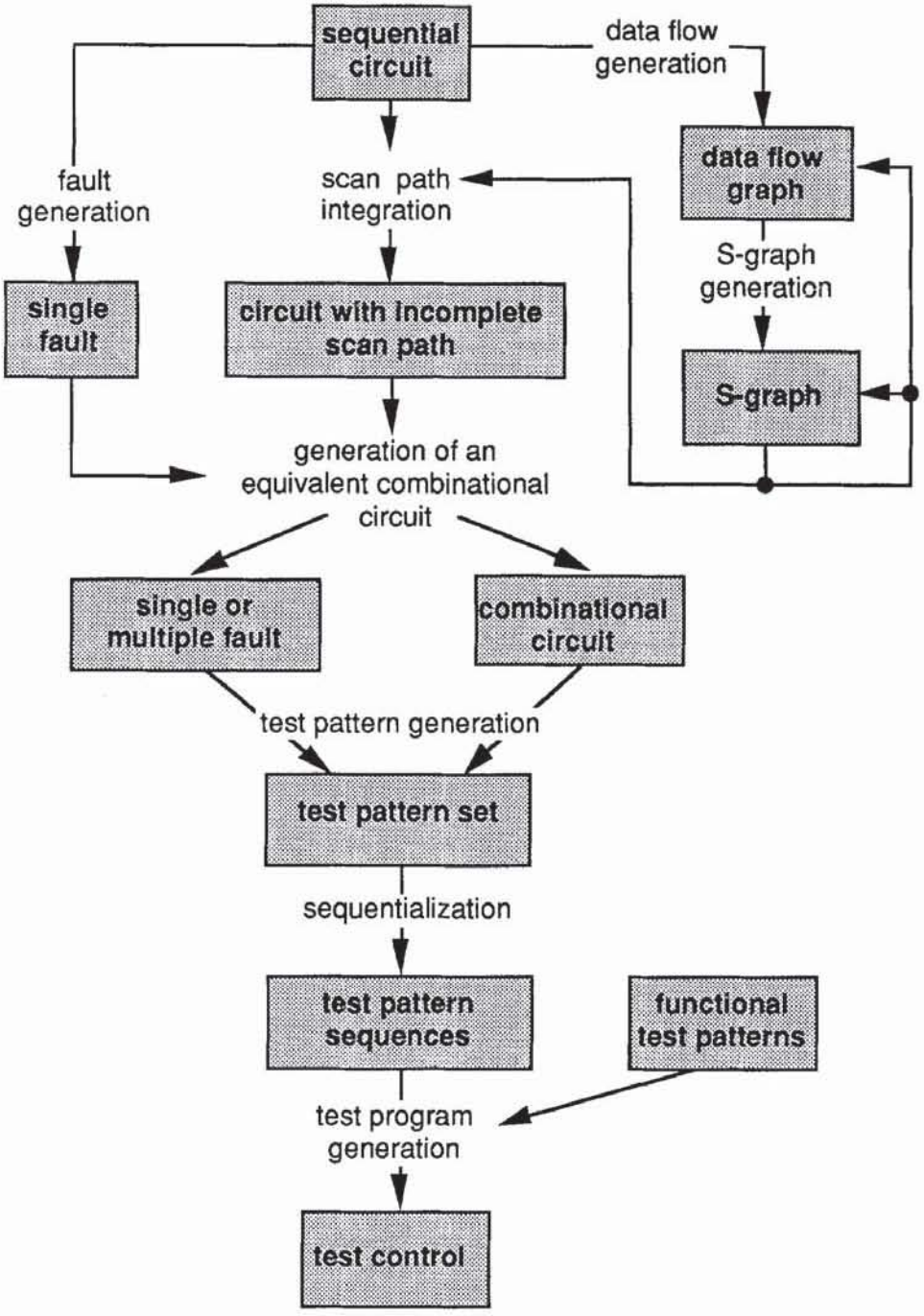

Fig. 12. Test system INSPIRATION.

For the general partial scan approach, only 17.2% and 20.6% of the flipflops must have the scan path capability. The multiplier (MU) has a structure, where generating an acyclic S-graph automatically provides an equidistant S-graph, too. The execution time for the sample circuits to determine the scan flipflops has always been less than 100 seconds (SUN 3/50).

It has already been mentioned that the test pattern generator must take advantage of the different scan techniques like acyclic or equidistant S-graphs. General purpose ATPG programs are not able to do so, as proved by table 4. The different circuit structures have been given to the program LASAR2000 [26], where the scan elements have been modeled as pseudo-primary inputs and outputs. Fault coverages obtained after 3600 seconds computing time are listed in table 3.

For the combinational parts of the circuits, the new test generator SPROUT-9V succeeded in identifying all

Table 4. Fault coverage by LASAR after 3600 seconds.

	AC	EQ	CS	Unmodified circuit
SP	13.2%	27.9%	80.7%	8.7%
MU	40.9%	40.9%	98.9%	9.8%
PP	47.9%	63.2%	93.0%	11.2%

redundancies. The number and percentage of redundant faults are listed in table 5. These faults have been removed from the fault list. By the presented approach, it is also possible to identify sequentially redundant faults requiring unreachable states. Different sets of states can be reached by an (AC)- and an (EQ)-design, and both sets contain the states reached during system operation. Table 6 gives the overall number of redundancies.

Table 5. Number and percentage of combinationally redundant faults.

	Number of combinationally redundant faults	Percentage
SP	4	0.1%
MU	0	0%
PP	188	7%

Table 6. Total number of redundancies.

	Total number of redundancies			Percentage		
	AC	EQ	CS	AC	EQ	CS
SP	4	4	4	0.1%	0.1%	0.1%
MU	10	10	0	0.4%	0.4%	0.0%
PP	692	512	188	25.6%	18.4%	7.0%

For the remaining faults, test patterns have been generated. The fault coverage with respect to all detectable faults and the necessary computing time are given in table 7. The time is measured on a workstation SUN 3/50.

Table 7. Fault coverage and computing time for different scan techniques by SPROUT-9V.

	AC		EQ		CS	
	Coverage	Time	Coverage	Time	Coverage	Time
SP	99.8%	1,099 sec	99.8%	1,411 sec	100%	238 sec
MU	100.0%	176 sec	100.0%	176 sec	100%	58 sec
PP	100.0%	1,101 sec	100.0%	1,060 sec	100%	513 sec

The rank of the S-graphs is in the range between 6 to 8. Compared with the size of the combinational part of the sequential circuits, the size of the combinational representations of the modified sample circuits grows in worst case only by 15%.

Up to now, the examples show that complete fault coverage is obtainable by scan paths containing only 20%–45% of all flipflops. Table 8 indicates the necessary test application times. We distinguish the number of shifting clocks (SH) and the number of system clocks (SY) for the different designs.

Table 8. Shifting clocks (SH) and system clocks (SY) for a complete test.

	AC		EQ		CS	
	SH	SY	SH	SY	SH	SY
SP	14,555	354	5,152	55	17,208	71
MU	6,696	92	6,696	92	16,653	90
PP	33,124	1182	12,660	210	38,760	284

The test lengths for equidistant S-graphs (EQ) are only a third of the test lengths for the more general acyclic graph (AC). On the other hand an acyclic graph needs only a half of the scan elements. This trade-off must be solved by the designer. Surprisingly, for all circuits the largest test time is needed using a complete scan path.

Conclusions

An efficient method has been proposed to select a minimal set of flipflops, which must be integrated into a scan path in order to guarantee testability. For the modified circuits, an ATPG-algorithm has been developed, providing complete fault coverage with respect to all irredundant faults. The proposed design and test method leads to lower hardware overhead and to shorter test times, thus reducing the overall test costs.

References

1. V.D. Agrawal, K. Cheng, D.D. Johnson, and T. Lin, "A complete solution to the partial scan problem," *Proc. IEEE Intern. Test Conf.*, pp. 44–51, September 1987.
2. V.D. Agrawal, K.-T. Cheng, D.D. Johnson, and T. Lin, "Designing circuits with partial scan, *IEEE Design & Test Comput.* 6 (2): 8–15, 1988.
3. M.J.Y. Williams and J.B. Angell, "Enhancing testability of large scale integrated circuits via test points and additional logic," *IEEE Trans. Comput.* C-22 (1), 1973.
4. R.G. Bennetts, *Design of Testable Logic Circuits*, Addison-Wesley, London, 1984.

5. J.J. LeBlanc, "LOCST: A built-in self-test technique," *IEEE Design & Test Comput.* 1 (4): 45–52, 1984.
6. K. Cheng and V.D. Agrawal, "An economical scan design for sequential logic test generation," *Proc. 19th Intern. Symp. Fault-Tolerant Comput.*, pp. 28–35, June 1989.
7. N. Christofides and S. Korman, "A computational survey of methods for the set covering problem, *Management Science* 21 (5): 591–599, 1975.
8. A. Ehrenfeucht, L. Fosdick, and L. Osterweil, "An algorithm for finding the elementary circuits of a directed graph," Tech. Rep. CU-CS-024-23, Dept. of Computer Science, University of Colorado, Boulder, 1973.
9. E.B. Eichelberger and T.W. Williams, "A logic design structure for LSI testability," *Proc. 14th ACM/IEEE Design Autom. Conf.*, pp. 462–468, 1977.
10. A.D. Friedman and P.R. Menon, *Theory and Design of Switching Circuits*, Computer Science Press, Rockville, MD, 1975.
11. H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *Proc. 13th Intern. Symp. Fault-Tolerant Comput.*, pp. 98–105, 1983.
12. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Francisco, 1979.
13. R. Gupta, R. Gupta, and M.A. Breuer, "BALLAST: A methodology for partial scan design," *Proc. 19th Intern. Symp. Fault-Tolerant Comput.*, pp. 118–125, 1989.
14. P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. Comput.* C-30 (3): 215–222, 1981.
15. L.H. Goldstein, "Controllability/observability analysis of digital circuits," *IEEE Trans. Circuits and Systems* CAS-26 (9): 685–693, 1979.
16. P. Gutberlet, "Entwurf eines schnellen Matrizen-Multiplizierers," Studienarbeit Fakultät Informatik, Universität Karlsruhe, 1988.
17. O. Haberl, "Entwurf und Implementierung eines PROLOG-Preprozessors als Standardzellen-Chip mit dem Entwurfssystem VENUS," Diplomarbeit an der Fakultät Informatik, Universität Karlsruhe, 1986.
18. J. Hartmanis, "Loop-free structure of sequential machines," *Information and Control* (5): 25–43, 1962.
19. D.B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM J. Comput.* 4 (1): 77–84, 1975.
20. R.M. Karp, "Reducibility among combinatorial problems," In R.E. Miller and J.W. Thatcher (eds.), *Complexity of Computer, Computations*, Plenum Press, New York, pp. 85–103, 1972.
21. A. Kunzmann, Verbesserung der Testbarkeit von Schaltwerken durch die Integration eines unvollständigen Prüfpfades," Interner Bericht 25/87, Universität Karlsruhe, Fakultät für Informatik, Oktober 1987.
22. A. Kunzmann, "Produktionstest mit deterministisch bestimmten Testmustern: der Testmustergenerator SPROUT," Interner Bericht 26/87, Universität Karlsruhe, Fakultät für Informatik, Oktober 1987.
23. A. Kunzmann, *Produktionstest synchroner Schaltwerke auf der Basis von Pipeline-Stukturen*, GI-Jahrestagung, Hamburg, 1988.
24. A. Kunzmann, "Test synchroner Schaltwerke auf des Basis partieller Prüfpfade," *VDI-Verlag, Fortschritt-Berichte* 10 (117), Düsseldorf, 1989.
25. A. Kunzmann and H.-J. Wunderlich, "Design automation of random testable circuits," *IIth European Solid State Circuits Conf.* (ESSCIRC), pp. 277–285, 1985.
26. *ETA LASAR Users Guide*, Vers. 4.7, November 1985.
27. H.T. Ma et al., "An incomplete scan design approach to test generation for sequential machines," *Proc. Intern. Test Conf.*, pp. 730–734, 1988.
28. P. Muth, "A nine-valued circuit model for test generation," *IEEE Trans. Comp.* C-25 (6): 630–636, 1976.
29. J.P. Roth, "Sequential test generation," *Technical Disclosure Bull.*, IBM, January 1978.
30. M.H. Schulz and E. Auth, "SOCRATES: Advanced automatic test pattern generation and redundancy identification techniques," *Proc. 18th Intern. Symp. Fault-tolerant Comput.*, pp. 30–35, Tokyo, 1988.
31. M.H. Schulz, E. Trischler, and T.M. Sarfert, "SOCRATES: A highly efficient automatic test pattern generation system," *Proc. Intern. Test Conf.*, 1987.
32. R. Tarjan, "Enumeration of the elementary circuits of a directed graph," *SIAM J. Comput.* 2 (3): 211–216, 1973.
33. J.C. Tiernan, "An efficient search algorithm to find the elementary circuits of a graph," *Comm. ACM*, 13, pp. 722–726, 1970.
34. E. Trischler, "Testability analysis and incomplete scan path," *Proc. Intern. Conf. Computer-Aided Design S*, pp. 38–39, 1983.
35. E. Trischler, "Incomplete scan path with an automatic test generation methodology," *Proc. Intern. Test Conf.*, pp. 153–162, 1984.
36. H. Weinblatt, "A new search algorithm for finding the simple cycles of a finite directed graph," *J. Assoc. Comput. Mach.*, 19: 43–56, 1972.
37. H.-J. Wunderlich, "PROTEST: A tool for probabilistic testability analysis," *Proc. 22th Design Autom. Conf.*, pp. 204–211, 1985.
38. H.-J. Wunderlich, "On computing optimized input probabilities for random tests," *Proc. 24th Design Autom. Conf.*, pp. 382–389, 1987.
39. H.-J. Wunderlich, "The design of random-testable sequential circuits," *Proc. 19th Intern. Symp. Fault-Tolerant Comput.*, pp. 110–111, 1989.

Arno B. Kunzmann received the diploma degree and the Dr. rer. nat. (Ph.D.) degree in computer science from the University of Karlsruhe, in 1982 and 1989, respectively.

Since 1983 he has been with the Institut for Computer Design and Fault Tolerance at the University of Karlsruhe, except the years 1985 and 1986, when he worked at the Computer Science Research Center Karlsruhe (FZI). During the first five years, his research was concentrated on the design and synthesis of integrated circuits. His current activities in research are concentrated on design for testability methods and algorithms for deterministic test pattern generation.

Hans-Joachim Wunderlich received the diploma degree in mathematics from the University of Freiburg, F.R. Germany, in 1981, and the Dr. rer. nat. (Ph.D.) degree from the University of Karlsruhe in 1986.

In 1982, he was a consultant at the Fraunhofer-Institute of Industrial Engineering, Stuttgart, where he worked in the field of operations research. In 1983, he joined the Institute of Computer Design and Fault Tolerance, University of Karlsruhe, where he has been the head of a research group on automation of circuit design and testing since 1986. His research interests include computer-aided design for testability, test generation, and digital simulation.