JOURNAL OF ELECTRONIC TESTING: Theory and Applications 17, 351–362, 2001

Application of Deterministic Logic BIST on Industrial Circuits

GUNDOLF KIEFER
Computer Architecture Lab, University of Stuttgart, Breitwiesenstr. 20/22, 70565 Stuttgart, Germany
Gundolf.Kiefer@informatik.uni-stuttgart.de

HARALD VRANKEN AND ERIK JAN MARINISSEN
Philips Research Laboratories, IC Design—Digital Design & Test, Prof. Holstlaan 4,
5656 AA Eindhoven, The Netherlands
Harald.Vranken@philips.com
Erik.Jan.Marinissen@philips.com

HANS-JOACHIM WUNDERLICH
Computer Architecture Lab, University of Stuttgart, Breitwiesenstr. 20/22, 70565 Stuttgart, Germany
wu@informatik.uni-stuttgart.de

Received August 1, 2000; Revised October 1, 2000

Editor: P. Prinetto

Abstract. We present the application of a deterministic logic BIST scheme based on bit-flipping on state-of-the-art industrial circuits. Experimental results show that complete fault coverage can be achieved for industrial circuits up to 100 K gates with 10,000 test patterns, at a total area cost for BIST hardware of typically 5–15%. It is demonstrated that a trade-off is possible between test quality, test time, and silicon area. In contrast to BIST schemes based on test point insertion no modifications of the circuit under test are required, complete fault efficiency is guaranteed, and the impact on the design process is minimized.

Keywords: logic BIST, industrial applications, scan-based BIST

1. Introduction

BIST (Built-In Self-Test) for random logic is becoming an attractive alternative in IC testing. Although research on logic BIST has been a hot topic already for over 30 years, the actual use of logic BIST in industry still is limited. However, recent advances in deep-submicron IC process technology and core-based IC design technology will certainly lead to more widespread use of logic BIST since external testing is becoming more and more difficult and costly [30]. This is confirmed by the ITRS (International Technology Roadmap for Semiconductors) statement that by 2014 it may cost more to test a transistor than to manufacture a transistor unless techniques like logic BIST are employed [23].

Modern system chips integrate digital logic, analog circuitry, and memory modules into a single IC. Testing such heterogeneous ICs requires either the use of multiple, dedicated testers or a single, complex tester for testing the digital, analog, and memory parts. The use of logic BIST for the digital logic modules, analog BIST for the analog modules, and memory BIST for the memory modules, provides that a single, low-cost external tester is sufficient [31].

Since more and more transistors are integrated on a single IC, the amount of test vectors necessary to test such large ICs is increasing. This requires large and costly vector memories in external test equipment. In addition, a significant increase is predicted for the number of transistors per pin, and consequently the amount of test data that has to be transported per pin will increase [23]. Combined with the growing gap between the internal clock frequencies and I/O frequencies, this gives rise to a bandwidth problem which will lead to increasing test times [30]. BIST may solve this problem since it provides on-chip generation of test stimuli and evaluation of test responses.

The growing need for performance-related testing also demands BIST solutions. Deep-submicron technologies introduce new performance-related defect types, and the increasing clock frequencies in high-speed designs impose aggressive timing margins. While the internal clock frequencies have risen by 30% per year, the accuracy of external test equipment has improved at a rate of only 12% per year [23]. Hence, it is becoming increasingly difficult to do performance-related testing using external test equipment. BIST may solve this problem since it allows to do accurate performance-related tests and precision measurement on-chip.

Logic BIST for industrial applications is currently supported by a few commercial CAD tools [15, 19] based on the STUMPS architecture for pseudo-random testing [4]. As shown in Fig. 1, the STUMPS scheme adds BIST hardware to a scannable circuit-under-test (CUT). Pseudo-random test stimuli are generated by a single linear-feedback shift register (LFSR), and the test responses are compressed by a multiple-input signature register (MISR). A test control unit controls the operation of the LFSR, MISR, and the scan chains in the CUT. If a single LFSR is used to generate bit sequences for multiple scan chains, these bit sequences only differ by a small number of shifts, and consequently the fault coverage may be reduced [3, 6]. For this reason, phase shifters [2, 5, 15, 21] are inserted between the LFSR and the scan chains in order to decorrelate the pseudo-random patterns. However, even decorrelated pseudo-random patterns cannot guarantee complete fault coverage.

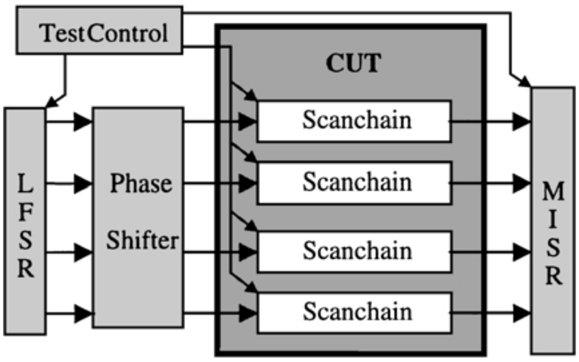

Fig. 1. STUMPS architecture with phase shifting logic.

The fault coverage can be improved by using an enhanced BIST scheme and/or by applying additional external test patterns. Commercial logic BIST tools currently offer to insert test points into the CUT in order to improve its pseudo-random testability [12, 22, 25]. Test point insertion however implies modifications of the CUT. Test points require additional silicon area and they may also have a negative impact on the timing behavior if inserted in critical paths. Timing verification after test point insertion is therefore required as an additional task in the design process, and subsequent design iterations may be required to solve timing violations. Furthermore, even test point insertion cannot guarantee complete fault coverage. For instance, 95–96% stuck-at fault coverage has been reported in [15] for the practical application of logic BIST with test point insertion on large industrial circuits (200–800 K gates).

A straightforward approach to achieve complete fault coverage is to apply additional external test patterns (i.e. 'top-off patterns' which have been generated by an ATPG tool) to the CUT. However, this approach still requires a considerable amount of external testing. In [7] it has been reported that detecting the last 10% of undetected faults typically requires 70% or more of the test patterns in an ATPG test set. In [15], external test patterns were applied on top of the BIST scheme with test points for improving the fault coverage from 95–96% to 96–97%, which still required 25–65% of the patterns in the full ATPG test set.

Complete fault coverage with neither CUT modifications nor external test patterns, can be achieved by using a BIST scheme containing a more sophisticated pattern generator that also addresses the random-pattern resistant faults. Examples of this approach are weighted random pattern generators [11, 24, 27, 28], pseudo-exhaustive pattern generators [1, 14], and deterministic pattern generators [13, 16–18, 26, 29]. However, the price for obtaining complete fault coverage usually is a relatively large amount of additional silicon area for the sophisticated pattern generator.

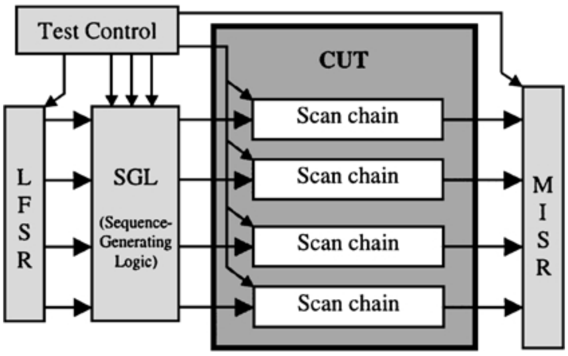

Fig. 2. Target structure of deterministic BIST scheme.

In [16, 17] a deterministic logic BIST scheme based on bit-flipping has been presented, which is more efficient than pseudo-random BIST in terms of silicon area and fault coverage for the ISCAS'85 and ISCAS'89 benchmark circuits [9, 10]. The basic principle of this BIST scheme is that the sequence of pseudo-random test patterns is modified by embedding deterministic patterns, as generated by an ATPG tool, into the sequence. The target structure of the deterministic logic BIST scheme is shown in Fig. 2. The pattern generator consists of an LFSR and a small combinational function, the sequence-generating logic (SGL). The SGL passes the bit sequences generated by the LFSR to the scan chains while modifying these sequences at certain bit positions which are selected by both the state of the test control unit and the state of the LFSR. These sequence modifications provide that additional deterministic patterns are generated and complete fault coverage can be guaranteed. The generated test patterns are applied to the CUT via the scan chains in a test-per-scan way.

In the present paper we describe the experiences obtained at Philips with the application of this bit-flipping logic BIST scheme on industrial designs. Logic BIST is currently not widely applied within Philips, but—like in most other semiconductor companies—many design groups are considering logic BIST as a solution to deal with growing test times, test pattern sizes, test frequencies, and test costs. In this context, we evaluated the applicability of bit-flipping logic BIST on industrial circuits. During the evaluation, researchers from the University of Stuttgart and Philips Research closely interacted, which resulted in a successful cross-fertilization.

Applying deterministic logic BIST in an industrial environment imposes various challenges. First, industrial circuits differ considerably from the ISCAS benchmark circuits, since they contain structures as tri-state buses, embedded memories, mixed-signal modules, and multiple clock domains. As for any logic BIST scheme, also deterministic logic BIST should consider these circuit structures, and bus contention and the propagation of unknown values into the MISR should be avoided. Second, the impact of deterministic logic BIST on the design process should be considered. Finally, the practical applicability of academic software tools in an industrial environment is often questionable since the requirements on CPU time and memory may be excessive with large industrial designs. In an academic environment, it may be well feasible to run a tool several times with different parameter settings on small circuits like the ISCAS circuits. However, doing the same for large industrial circuits in an industrial environment may be prohibitive.

In the remainder of this paper, first our BIST synthesis algorithm is summarized in Section 2. The evaluation method is outlined in Section 3, in which the industrial circuits and their preparations for logic BIST, as well as practical tool issues and impact on the design process are discussed. Finally, experimental results are presented in Section 4, and Section 5 concludes the paper.

2. BIST Synthesis Algorithm

The synthesis algorithm for the sequence-generating logic (SGL) is described in [17, 18, 29]. For a better understanding of the rest of the paper, a brief summary follows.

The structure of the SGL is shown in Fig. 3. It consists of XOR gates and a bit-flipping function (BFF). The current state of the LFSR as well as the bit counter and the pattern counter of the test control unit serve as inputs for the BFF.

Assume the BFF is empty (i.e. all outputs being 0), so that the structure in Fig. 3 is equivalent to a simple LFSR. This LFSR generates a sequence of pseudo-random patterns which may detect the majority (e.g. 90%) of all faults. However, many of these patterns do not contribute to the fault coverage, since they only detect faults which already are detected by other patterns in the sequence. These "useless" patterns are nevertheless required to bring the LFSR from one useful pattern to another useful pattern. Hence, the useless patterns may be modified without reducing the fault coverage.

The BIST synthesis procedure determines these useless patterns and runs an automatic test pattern

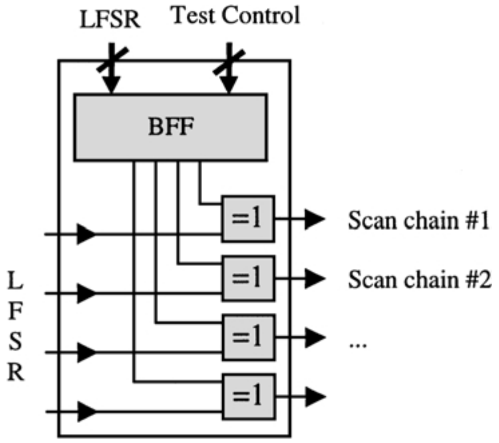

Fig. 3. Sequence-generating logic.

generation (ATPG) tool for generating deterministic patterns for the untested faults. The BFF is constructed such that some of the useless patterns are turned into deterministic patterns so that the pattern generator provides sufficiently high (e.g. complete) fault coverage. In [29] it has been observed that for this purpose the BFF only has to modify a relatively small number of bits and thus can be very small.

In order to achieve an efficient implementation of the pattern generator, the BFF is constructed by means of an iterative algorithm as sketched in Fig. 4. The algorithm starts with an empty BFF and terminates when sufficient fault coverage is achieved. In each iteration, the BFF is enhanced, so that new deterministic patterns are produced while certain old patterns, which are essential for detecting faults that are already detected, remain unchanged.

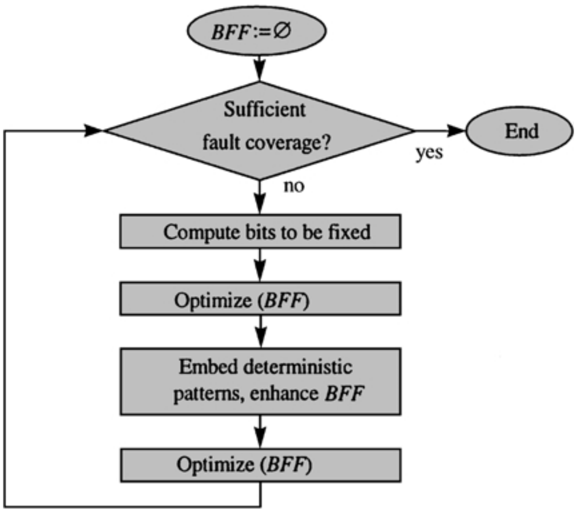
BFF:=∅
Sufficient fault coverage?
yes
End
no
Compute bits to be fixed
Optimize (BFF)
Embed deterministic patterns, enhance BFF
Optimize (BFF)

Fig. 4. BFF synthesis algorithm.

Pattern bits not to be changed are determined by three-valued fault simulation. The states of the LFSR and the test control unit which correspond to these essential bits, specify inputs for the BFF for which its function must not change.

For all the currently undetected faults, deterministic patterns are computed by an ATPG tool. In order to obtain a small BFF, the ATPG tool should minimize the number of specified bits, e.g. as described in [13]. For one or several of these ATPG patterns, patterns from the LFSR and current BFF with a minimum number of conflicting bits are selected. The ATPG pattern(s) are then embedded by enhancing the BFF in such a way that the conflicting bits are modified.

Most bits generated by the on-chip pattern generator correspond neither to relevant pseudo-random nor to embedded deterministic patterns and thus specify don't-care conditions which can be exploited for minimizing the BFF. For this purpose ESPRESSO-like logic optimization procedures [8] are integrated into the main loop of the algorithm (see Fig. 4).

Our BIST scheme does not require any additional phase shifting logic [17]. Furthermore, since the outputs of the pattern generator depend on the current state of the test control unit, the LFSR may be very small. This does not only reduce the area of the LFSR, but also the BFF may be smaller as the autocorrelation of deterministic test patterns can be exploited [16].

The synthesis procedure is not limited to single deterministic patterns (e.g. for stuck-at testing in full scan designs), but can also handle pattern sequences as required for partial scan designs or delay fault testing [18].

3. Evaluation Method

3.1. *Industrial Circuits*

In general, industrial circuits differ considerably from the ISCAS'85 and ISCAS'89 [9, 10] benchmark circuits. The ISCAS circuits are older designs originating from the 1980's, and they are relatively small: there are only 5 circuits that have more than 10,000 signal lines and the largest circuit (s38584) contains 38,584 signal lines. Furthermore, the ISCAS circuits contain logic gates only and they do not include bus structures, embedded memories, mixed-signal modules, and multiple clock domains as typically found in industrial circuits.

We applied the deterministic logic BIST scheme on 30 state-of-the-art industrial circuits from various application fields. These industrial circuits are modules extracted from various Philips designs, including an 8-bit microcontroller, a programmable DSP, and various dedicated ICs for video processing, audio processing, graphics processing, and telecommunication. All circuits have been proven on silicon. They are all full-scan designs with muxed-scan flipflops configured in multiple scan chains, and some circuits contain embedded memories and multiple clock domains. Table 1 shows the complexity of the circuits in terms of number of signal nets, NAND-equivalent gates (i.e. circuit area divided by the area for a 2-input NAND gate), flipflops (FF), primary inputs (PI), and primary outputs (PO). Currently, our bit-flipping BIST tool imposes certain requirements on memory and CPU time such that only circuits up to 100 K gates can be handled in practice. However, these industrial circuits are still larger than the ISCAS circuits.

In order to prepare the circuits for logic BIST, we manually made the following modifications to the circuits. These or similar modifications are required for any logic BIST scheme in order to be able to access the complete circuit and to avoid the propagation of unknown values into the MISR. We assumed that these circuit modifications did not corrupt the timing behavior, but we did not verify this assumption.

a) We provided that a single clock is used in multiple clocks circuits when the BIST logic is active. Hence, all clock domains run on the same frequency in test mode.
b) We modified the embedded memories either by adding a scan chain around these embedded memories or by adding bypass logic. In case of a surrounding scan chain, all memory inputs and outputs can be controlled and observed directly by flipflops. In case of bypass logic, the memory inputs are connected directly to the memory outputs by means of multiplexers and XOR-gates in test mode.
c) We added a scan chain around each circuit, which allows all primary inputs to be controlled from the LFSR and all primary outputs to propagate into the MISR.

Table 1. Circuit characteristics.

Circuit	#nets	#gates	#FF	#PI	#PO
p2221	2,221	2,208	217	69	19
p2441	2,441	2,270	183	68	95
p2675	2,675	1,788	108	274	128
p4210	4,210	4,218	305	62	52
p4250	4,250	3,597	184	13	76
p4919	4,919	4,779	358	80	77
p5918	5,918	4,444	201	107	49
p6291	6,291	5,976	382	85	39
p7318	7,318	8,832	1,345	8	15
p7890	7,890	6,771	381	64	77
p8689	8,689	8,039	500	59	70
p8873	8,873	10,722	476	85	45
p9041	9,041	10,246	369	289	33
p10705	10,705	10,355	635	123	120
p12292	12,292	9,957	365	203	67
p13033	13,033	12,314	711	251	92
p13651	13,651	11,434	582	134	38
p14473	14,473	12,356	690	194	446
p17828	17,828	19,323	1,731	124	134
p22383	22,383	25,834	966	631	144
p23572	23,572	28,692	1,369	131	145
p24370	24,370	20,448	683	325	164
p25015	25,015	22,848	1,466	381	69
p27369	27,369	20,991	1,134	253	115
p27530	27,530	20,089	400	159	65
p44177	44,177	49,073	3,644	72	71
p52251	52,251	54,914	4,676	144	252
p52922	52,922	69,986	3,922	179	2,486
p64984	64,984	63,508	5,593	53	49
p80590	80,590	92,319	2,675	1,025	159

A circuit may also be prepared for logic BIST using more advanced techniques, such as using a BIST controller to support multiple clocks during test mode [15, 20], adding control and observation points to avoid the propagation of unknown values, initialization of embedded memories with known values before activating BIST (e.g. by running memory BIST first), using memory pass-through mode in which the same memory address is written and read simultaneously, controlling primary inputs by means of mux-type control points driven from nearby flipflops, and observing primary outputs by means of observation points connected to nearby flipflops [15].

3.2. *Evaluation Flow and Tool Issues*

The target in our initial experiments was to apply deterministic logic BIST for achieving 100% fault efficiency (i.e. all non-redundant faults will be detected) using 10,000 pseudo-random test patterns. During these experiments we varied parameters as the LFSR size and the LFSR polynomial. In later experiments, we explored trade-offs in test quality, test time, and silicon area by also varying the number of test patterns and the target fault efficiency.

The application of deterministic logic BIST is currently supported by an academic software package, in which the tool for the generation of the sequence-generating logic is most relevant. In addition, the academic software package provides tools for netlist compilation, scan chain identification, fault simulation, and ATPG. As expected, these academic software tools are not as advanced as professional, commercial tools. Software limitations and requirements on memory and CPU time cause that only circuits up to 100 K gates can be handled practically. However it is expected that these limitations can be removed when linking the tool for generating the sequence-generating logic to professional CAD tools.

The flow used during the evaluation is depicted in Fig. 5. We extracted circuits from large Philips IC designs, and manually prepared these circuits for logic BIST as outlined in the previous section. Subsequently, we applied Philips inhouse tools for DfT rule checking and design rule checking. We performed ATPG to obtain figures on compact test set sizes and fault coverage. Finally, we ran the BIST synthesis tool to generate the BIST hardware.

The output of the BIST synthesis tool is a register-transfer level VHDL description of the complete BIST hardware, including the LFSR, the MISR, the BIST controller, and the sequence-generating logic. In addition, the BIST tool provides a logfile with detailed information on the embedded deterministic patterns, fault efficiency, and fault coverage. The VHDL code of the BIST hardware can be synthesized using a state-of-the-art logic synthesis tool and can be mapped to an arbitrary technology library, allowing optimization for area or timing. For the evaluation we used Synopsys' Design Compiler and a Philips 0.25 μm CMOS technology library. The final steps would be to assemble the CUT and the BIST hardware and to perform placement and routing as well as timing analysis, however we did not perform these final steps.

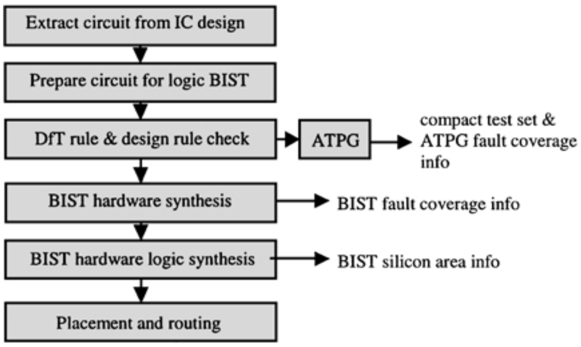
Extract circuit from IC design
Prepare circuit for logic BIST
DfT rule & design rule check
ATPG
compact test set & ATPG fault coverage info
BIST hardware synthesis
BIST fault coverage info
BIST hardware logic synthesis
BIST silicon area info
Placement and routing

Fig. 5. Evaluation flow.

3.3. Impact on the Design Process

Applying deterministic logic BIST in an industrial environment implies that additional tasks have to be performed in the design process. The first task is preparing the CUT for logic BIST to avoid bus contention and the propagation of unknown values into the MISR. For existing circuits, these preparations imply the modification of the circuit as an additional step in the design process as well as verification steps. For circuits to be designed from scratch, the preparations for logic BIST can be considered during the design process, and hence bus contention and propagation of unknowns into the MISR can be avoided in advance.

Like in most other BIST approaches, our BIST synthesis tool operates on a gate-level netlist of the CUT. Hence, the generation of the BIST hardware is done relatively late in the design process. An advantage of deterministic logic BIST is that no further modifications of the CUT are required at this point. This is in contrast to BIST schemes based on test point insertion, which imply modification of the CUT and additional design steps to verify timing constraints which may be followed by another synthesis iteration in order to fix timing violations. Hence, deterministic logic BIST has less impact on the design process than pseudo-random BIST schemes using test point insertion.

4. Experimental Results

4.1. Fault Coverage and Fault Efficiency

Table 2 shows the fault coverage (i.e. number of detected faults with respect to all faults) obtained with an external deterministic pattern set (FC_{ATPG}). By definition, deterministic pattern sets always provide complete coverage of all testable faults (fault efficiency $FE_{ATPG} = 100\%$), and thus FC_{ATPG} indicates the percentage of untestable faults in the respective circuits.

Table 2. Fault coverage and fault efficiency.

Circuit	FC_{ATPG}	FC_{Random}	FE_{Random}
p2221	96.83%	79.29%	81.88%
p2441	97.73%	83.99%	85.94%
p2675	99.91%	99.45%	99.54%
p4210	99.99%	94.46%	94.47%
p4250	99.64%	95.74%	96.09%
p4919	98.91%	96.00%	97.06%
p5918	98.60%	95.18%	96.53%
p6291	99.22%	96.04%	96.80%
p7318	99.52%	99.37%	99.84%
p7890	99.67%	98.62%	98.95%
p8689	99.98%	88.54%	88.56%
p8873	97.89%	94.36%	96.40%
p9041	99.97%	96.97%	97.00%
p10705	99.79%	95.15%	95.35%
p12292	99.49%	96.72%	97.21%
p13033	99.52%	95.11%	95.57%
p13651	97.66%	97.50%	99.84%
p14473	98.99%	85.52%	86.39%
p17828	99.03%	95.88%	96.82%
p22383	99.84%	92.85%	92.99%
p23572	94.11%	80.50%	85.53%
p24370	98.60%	93.08%	94.40%
p25015	95.52%	93.89%	98.30%
p27369	96.24%	95.05%	98.77%
p27530	99.91%	98.14%	98.23%
p44177	98.89%	96.17%	97.25%
p52251	99.60%	98.92%	99.32%
p52922	96.93%	89.81%	92.66%
p64984	99.08%	93.20%	94.07%
p80590	99.58%	98.68%	99.09%

We applied 10,000 pseudo-random test patterns to all circuits. Table 2 shows the fault coverage and the fault efficiency (i.e. number of detected faults with respect to non-redundant faults) that is obtained with these pseudo-random test patterns (FC_{Random} and FE_{Random}).

For 20 out of the 30 circuits the random fault efficiency (FE_{Random}) is better than 95%. Hence in situations where limited fault efficiency is sufficient, such as field testing, pseudo-random BIST already may be enough. However, Table 2 also shows that 100% fault efficiency is never achieved with pseudo-random patterns only. Hence, all these circuits require deterministic logic BIST to improve fault efficiency to the level as typically required for production testing.

When applying deterministic logic BIST, 100% fault efficiency was obtained for 15 out of the 30 circuits. For the remaining 15 circuits we obtained a fault efficiency >99.9%. The reason for not achieving 100% fault efficiency in all cases is that our academic ATPG tool aborted on some faults. Hence, not reaching 100% fault efficiency is purely due to tool limitations and not at all a limitation of the method as such. Since the number of aborted faults is less than 0.1% in all cases, their impact on the synthesis results reported below can be considered neglectable.

4.2. Deterministic Test Patterns

The efficiency of the deterministic BIST scheme strongly depends on the number of specified bits in deterministic patterns. Table 3 provides figures on the deterministic patterns for the industrial circuits: the number of embedded deterministic patterns (#det. pats), the number of pseudo-primary inputs (#PPI), and the maximum number (#spec. bits max.), the minimum number (#spec. bits min.), and the average number (#spec. bits aver.) of specified bits in the embedded deterministic patterns.

Table 3 indicates that the number of specified bits is quite small and does generally not increase with the size of the circuit or the number of pseudo-primary inputs. A similar observation has been reported with the ISCAS'85 and ISCAS'89 benchmark circuits in [13]. For circuits which require a relatively large maximum number of specified bits (e.g. p24370 and p64984), only few patterns actually require this maximum number. As a consequence, good encodability of deterministic patterns can be expected even for circuits larger than those presented in this paper.

4.3. Silicon Area

To estimate the silicon area, we synthesized the circuits using Synopsys' Design Compiler, using optimization for area, and a Philips 0.25 μm CMOS technology library. The results are shown in Table 4. The numbers in Table 4 are estimates of the actual silicon area since they only include the area for logic gates while leaving out the area for wiring as well as the area for embedded memories. Table 4 shows the total silicon area for the scannable circuit (core area), the percentage of silicon area required for making the circuit fully scannable (scan area), and the percentage of silicon area required for the complete BIST hardware (BIST area).

Table 3. Embedded deterministic patterns.

			#Spec. bits		
Circuit	#Det. pats	#PPI	Max.	Min.	Aver.
p2221	61	286	25	14	20.4
p2441	48	251	30	11	23.8
p2675	3	382	14	8	10.0
p4210	61	367	122	49	68.8
p4250	108	197	35	12	18.4
p4919	78	438	26	12	17.9
p5918	61	308	25	16	21.0
p6291	128	467	109	9	49.6
p7318	4	1,353	12	8	10.3
p7890	43	445	36	17	24.7
p8689	490	559	34	13	21.5
p8873	118	561	68	8	51.4
p9041	139	658	58	19	37.7
p10705	179	758	35	11	22.7
p12292	173	568	78	16	38.4
p13033	275	962	47	12	22.6
p13651	3	716	39	32	36.3
p14473	595	884	111	14	40.8
p17828	289	1,855	31	11	19.9
p22383	654	1,597	79	13	39.2
p23572	1,025	1,500	35	12	20.9
p24370	595	1,008	103	13	32.5
p25015	190	1,847	79	12	27.1
p27369	106	1.387	57	10	22.3
p27530	309	559	93	16	41.1
p44177	681	3,716	55	10	18.4
p52251	135	4,813	28	11	17.1
p52922	995	4,101	80	4	36.7
p64984	2,142	5,646	364	7	27.8
p80590	356	3,700	51	13	30.2

Table 4. Silicon area.

			BIST area		
Circuit	Core area (μm^2)	Scan area	Total	LFSR, MISR, & test control	SGL
p2221	59,607	16.38%	26.91%	17.41%	9.50%
p2441	61,281	13.44%	25.50%	16.55%	8.94%
p2675	48,267	10.07%	24.89%	24.41%	0.48%
p4210	113,877	12.05%	22.60%	10.58%	12.01%
p4250	97,110	8.53%	20.87%	11.95%	8.92%
p4919	129,033	12.49%	13.75%	9.29%	4.46%
p5918	119,979	7.54%	16.09%	10.07%	6.02%
p6291	161,352	10.65%	14.75%	7.50%	7.25%
p7318	238,464	25.42%	4.89%	4.82%	0.07%
p7890	182,826	9.38%	10.26%	6.61%	3.65%
p8689	217,053	10.37%	25.41%	5.69%	19.72%
p8873	289,053	7.40%	10.27%	4.14%	6.12%
p9041	276,642	6.00%	16.57%	4.51%	12.06%
p10705	279,594	10.27%	11.36%	4.42%	6.93%
p12292	268,830	6.11%	20.08%	4.64%	15.43%
p13033	332,478	9.62%	11.59%	3.67%	7.92%
p13651	308,727	8.48%	4.18%	3.91%	0.27%
p14473	333,612	9.31%	34.15%	3.87%	30.27%
p17828	521,712	14.93%	6.75%	2.40%	4.34%
p22383	697,527	6.23%	20.90%	1.97%	18.93%
p23572	774,684	7.95%	10.34%	1.64%	8.70%
p24370	552,096	5.57%	19.61%	2.92%	16.69%
p25015	616,887	10.69%	5.93%	1.99%	3.95%
p27369	566,748	9.00%	4.29%	2.25%	2.04%
p27530	542,394	3.32%	15.36%	2.32%	13.04%
p44177	1,324,980	12.38%	4.42%	0.95%	3.47%
p52251	1,482,687	14.19%	1.50%	0.88%	0.61%
p52922	1,889,622	9.34%	8.42%	0.86%	7.57%
p64984	1,714,725	14.71%	11.26%	0.80%	10.45%
p80590	2,492,622	4.94%	2.91%	0.58%	2.33%

The silicon area for the complete BIST hardware is the sum of the silicon area for the LFSR, MISR, test controller, and the sequence-generating logic (SGL). All percentages in Table 4 are relative to the area of the full-scannable circuit. The same information is depicted in Fig. 6.

The silicon area for the LFSR, MISR, and test controller can be considered as the lower bound on silicon area for pseudo-random BIST. Since we applied 10,000 pseudo-random patterns to all circuits, the size of the required LFSR, MISR, and test controller is almost constant for all circuits. Hence, as shown in Table 4, the percentage of silicon area for these components decreases with increasing circuit size to less than 1% for the largest circuits. For larger circuits, the sequence-generating logic tends to be the major part of the BIST hardware.

Table 4 shows that for circuits larger than 5 K gates, the total silicon area for BIST with 10,000 patterns typically is 5–15%. Of the 23 circuits larger than 5 K gates, 6 circuits require less than 5% silicon area, 10 circuits require 5–15% silicon area, and the 7 remaining circuits require more than 15% silicon area. In addition, 3 of the 5 largest circuits require less than 5% silicon.

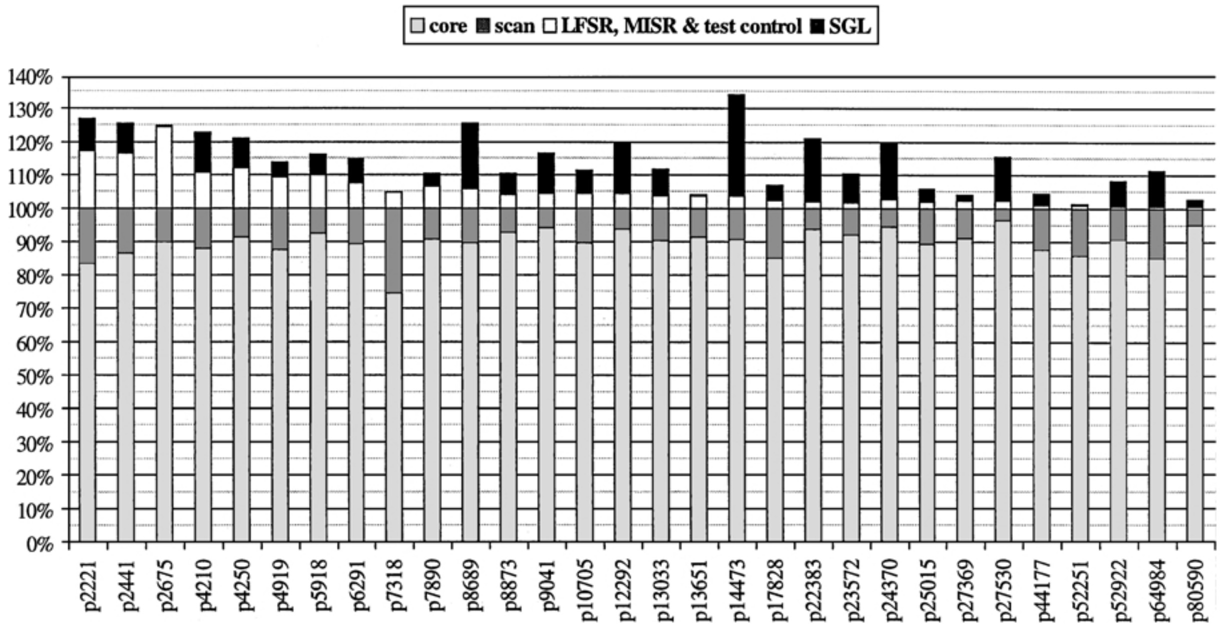

Fig. 6. Silicon area (10,000 patterns, 100% fault efficiency).

The size of the sequence-generating logic primarily depends on the random-testability of the circuit. Circuits that are badly random testable require a larger SGL than circuits that are well random testable. For instance, circuit p13651 is well random testable (99.84% $\text{FE}_{\text{Random}}$) and the total BIST hardware requires 4.18% silicon area, while circuit p14473 is badly random testable (86.39% $\text{FE}_{\text{Random}}$) and the total BIST hardware requires 34.15% silicon area. One way to decrease the required silicon area is to improve the random testability of the circuit, e.g. by test point insertion. However this requires modifications of the CUT and increases the silicon area of the CUT.

Looking at the scan area numbers in Table 4, it is noticeable that up to 25% of the chip area (or up to 15% when only considering the large modules with more than 10 K gates) is spent on scan. In several cases, e.g. for the 5 largest circuits with more than ~50 K gates, the total area for deterministic BIST is less than the area for the scan design.

4.4. *Trade-Offs*

A trade-off is possible between test time and silicon area: when increasing the number of patterns, the test time will increase but the silicon area for the BIST hardware will decrease. In all experiments so far, we applied 10,000 test patterns, which is relatively few. Increasing the number of test patterns will provide higher random fault efficiency so that (1) less deterministic patterns have to be embedded, and in addition (2) the degree of freedom for embedding deterministic patterns is enhanced. Both effects cause that the silicon area for the sequence-generating logic decreases. Applying more test patterns requires a larger size of the test controller, but this increase is only marginal.

These effects are clearly demonstrated in Table 5, which shows for a number of circuits how the silicon area for BIST decreases if more test patterns are applied. For instance, increasing the number of patterns for circuit p9041 from 10 K to 256 K results in 4.72% reduction of silicon area. For this circuit, the random fault efficiency improves when increasing the number of patterns from 10 K to 64 K, and hence in this range both effects (1) and (2) are present. When applying more than 64 K patterns, the random fault efficiency does not improve any further, but the silicon area still decreases due to larger freedom for embedding deterministic patterns. Similar observations can be made for circuit p14473 and p23572. In all cases reported in Table 5, the fault efficiency obtained with deterministic BIST is >99.9%.

Another trade-off is possible between test quality and silicon area: when lowering the target fault efficiency, the silicon area for BIST hardware will decrease. It is generally known that the major part of

Table 5. Trade-off between test time and silicon area.

			BIST silicon area		
Circuit	#Test patterns	$\mathrm{FE}_{\mathrm{Rand}}$	Total	LFSR, MISR, & test control	SGL
p9041	10,000	97.00%	16.57%	4.51%	12.06%
	32,768	97.54%	14.50%	4.67%	9.83%
	65,536	98.24%	13.83%	4.88%	8.95%
	131,072	98.24%	12.63%	4.98%	7.65%
	262,144	98.24%	11.85%	5.09%	6.75%
p14473	10,000	86.39%	34.15%	3.87%	30.27%
	65,536	91.01%	30.27%	4.20%	26.07%
	131,072	91.01%	28.16%	4.24%	23.92%
	262,144	91.01%	28.09%	4.32%	23.78%
p23572	10,000	85.53%	10.34%	1.64%	8.70%
	65,536	95.42%	5.60%	1.83%	3.76%
	131,072	95.42%	5.15%	2.11%	3.04%
p24370	10,000	94.40%	19.61%	2.92%	16.69%
	131,072	97.09%	11.74%	3.10%	8.64%
p27530	10,000	98.23%	15.36%	2.32%	13.04%
	32,768	98.99%	11.49%	2.36%	9.12%
	65,536	99.08%	9.57%	2.47%	7.10%
p44177	10,000	97.25%	4.42%	0.95%	3.47%
	131,072	99.18%	2.49%	1.22%	1.27%

a test set is required for detecting the last few percents of undetected faults. For instance, in [7] it is reported that 70% or more of a full ATPG test set is required for detecting the last 10% of undetected faults, and in [15] it is reported that 25–65% of a full ATPG test set is required for improving fault coverage from 95–96% to 96–97%. A similar observation can be made for deterministic logic BIST: most silicon area in the sequence-generating logic is required for detecting the last remaining undetected faults.

This is clearly demonstrated in Table 6 for four circuits. For instance, decreasing the target fault efficiency for circuit p9041 from 100% to 99% results in 6.16% reduction of silicon area, while even 9.06% reduction is achieved for circuit p14473. Fig. 7 shows the trade-offs between test time, test quality, and silicon area for circuit p9041 in a single graph. The above experiments indicate that deterministic logic BIST enables to make a trade-off between test quality (i.e. fault efficiency), test time (i.e. number of test patterns), and silicon area. This is attractive in an industrial environment in which different trade-offs can be made for different types of products.

Table 6. Trade-off between test quality and silicon area.

		BIST silicon area		
Circuit	$\mathrm{FE}_{\mathrm{BIST}}$	Total	LFSR, MISR, & test control	SGL
p9041	100%	16.57%	4.51%	12.06%
	99.00%	10.41%	4.51%	5.90%
	98.00%	7.03%	4.51%	2.52%
	97.00%	4.51%	4.51%	0%
p14473	100%	34.15%	3.87%	30.27%
	99.00%	25.09%	3.80%	21.29%
	96.00%	16.21%	3.75%	12.46%
	86.39%	3.87%	3.87%	0%
P23572	100%	10.34%	1.64%	8.70%
	98.00%	5.83%	1.64%	4.19%
	85.53%	1.64%	1.64%	0%
P52922	100%	8.42%	0.86%	7.57%
	99.00%	7.59%	0.86%	6.74%
	96.00%	3.09%	0.84%	2.25%
	92.66%	0.86%	0.86%	0%

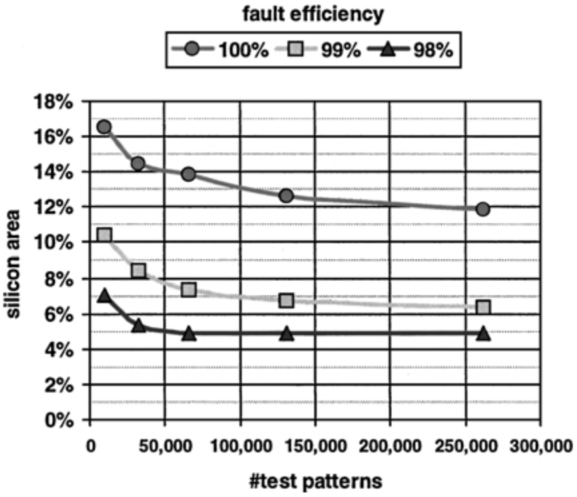

Fig. 7. Trade-offs for circuit p9041.

5. Conclusion

We presented the application of a deterministic logic BIST scheme based on bit-flipping on industrial designs. Our experimental results show that complete fault efficiency can be achieved for circuits up to 100 K gates with 10,000 test patterns at a total area cost for BIST hardware of typically 5–15%. The size of the

BIST hardware primarily depends on the random testability of the CUT.

We also demonstrated that a trade-off can be made between test time, test quality, and silicon area. For circuits that are reasonably well random testable, the size of the BIST hardware may be reduced to typically less than 10% by increasing the test time (i.e. the number of test patterns). Decreasing the target fault efficiency from 100% to 99% may result in a considerable reduction of the silicon area for the sequence-generating logic.

As any STUMPS-based BIST scheme, bit-flipping logic BIST requires that the CUT is a BIST-ready scan design. However, the bit-flipping BIST scheme does not require additional modifications of the CUT, which is in contrast to BIST schemes based on test point insertion. Hence, the impact on the system behavior and the design process is minimized. In addition, bit-flipping logic BIST guarantees that the target fault efficiency is reached.

Our overall conclusion is that the application of bit-flipping logic BIST on industrial circuits is very well feasible. It is expected that the results reported in this paper will hold even for larger circuits, although further experiments are required to confirm this.

In future work, we intend to consider deterministic logic BIST for other fault models than stuck-at, in particular delay faults. This implies both determining how well other faults are covered by the current deterministic BIST scheme for stuck-at faults, as well as experiments on embedding pattern sequences for detecting delay faults. We are currently improving our deterministic BIST tools, which should enable us to handle circuits much larger than 100 K gates.

6. Acknowledgment

Part of this work has been supported by the DFG under grants Wu 245/1-1 and Wu 245/1-3. We acknowledge all Philips design groups which provided netlists and data for the circuits used in this paper, in particular we thank Chris Feige and Jan ten Pierick. We also acknowledge the team of Friedrich Hapke at the Philips Software Development Center in Hamburg for the discussions on tool integration.

References

1. S.B. Akers, "On the Use of Linear Sums in Exhaustive Testing," in *Proceedings International Symposium on Fault-Tolerant Computing*, IEEE, 1985, pp. 148–153.
2. P.H. Bardell, "Design Considerations for Parallel Pseudo-random Pattern Generators," *Journal of Electronic Testing: Theory and Applications (JETTA)*, Vol. 1, No. 1, pp. 73–87, 1990.
3. P.H. Bardell, "Calculating the Effects of Linear Dependencies in m-Sequences Used as Test Stimuli," *IEEE Transactions on CAD*, pp. 83–86, Jan. 1992.
4. P.H. Bardell and W.H. McAnney, "Parallel Pseudo-random Sequences for Built-in Test," in *Proceedings International Test Conference*, IEEE, 1984, pp. 302–308.
5. P.H. Bardell and W.H. McAnney, "Pseudo-random Arrays for Built-in Tests," *IEEE Transactions on Computers*, Vol. C-35, No. 7, pp. 653–658, 1986.
6. P. Bardell, W.H. McAnney, and J. Savir, *Built-in Test for VLSI*, New York: Wiley-Interscience, 1987.
7. R.W. Bassett, B.J. Butkus, S.L. Dingle, M.R. Faucher, P.S. Gillis, J.H. Panner, J.G. Petrovik, and D.L. Weather, "Low Cost Testing of High Density Logic Components," in *Proceedings International Test Conference*, IEEE, 1989, pp. 550–557.
8. R.K. Brayton, G.D. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Boston: Kluwer Academic Publishers, 1984.
9. F. Brglez, D. Bryan and K. Komzminski, "Combinational Profiles of Sequential Benchmark Circuits," in *Proceedings International Symposium on Circuits and Systems*, IEEE, 1989, pp. 1929–1934.
10. F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *Proceedings International Symposium On Circuits and Systems*, IEEE, 1985, pp. 663–698.
11. F. Brglez, C. Gloster, and G. Kedem, "Hardware-Based Weighted Random Pattern Generation for Boundary-Scan," in *Proceedings International Test Conference*, IEEE, 1989, pp. 264–274.
12. J.P. Hayes and A.D. Friedman, "Test Point Placement to Simplify Fault Detection," *IEEE Transactions on Computers*, Vol. C-33, pp. 727–735, July 1974.
13. S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern Generation for a Deterministic BIST Scheme," in *Proceedings International Conference on Computer-Aided Design*, IEEE, 1995, pp. 88–94.
14. S. Hellebrand, H.-J. Wunderlich, and O.F. Haberl, "Generating Pseudo-Exhaustive Vectors for External Testing," in *Proceedings International Test Conference*, IEEE, 1990, pp. 670–679.
15. G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, "Logic BIST for Large Industrial Designs: Real Issues and Case Studies," in *Proceedings International Test Conference*, IEEE, 1999, pp. 358–367.
16. G. Kiefer and H.-J. Wunderlich, "Using BIST Control for Pattern Generation," in *Proceedings International Test Conference*, IEEE, 1997, pp. 347–355.
17. G. Kiefer and H.-J. Wunderlich, "Deterministic BIST with Multiple Scan Chains," in *Proceedings International Test Conference*, IEEE, 1998, pp. 1057–1064.
18. G. Kiefer and H.-J. Wunderlich, "Deterministic BIST with Partial Scan," in *Proceedings European Test Workshop*, IEEE, 1999, pp. 110–116.
19. S. Mukherji, L. Nguyen, D. Burek, and S. Baird, "IP/VC-Based Test Methodology (Part-1): A Case Study," *Digest of Papers*

International Workshop on Testing Embedded Core-Based Systems, IEEE, paper 1.2, pp. 1.9, 1998.

20. B. Nadeau-Dostie, D. Burek, and A. Hassan, "Scan-BIST: A Multifrequency Scan-Based BIST Method," *IEEE Design & Test of Computers*, Vol. 11, No. 1, pp. 7–17, 1994.
21. J. Rajski, N. Tamarapalli, and J. Tyszer, "Automated Synthesis of Large Phase Shifters for Built-In Self-Test," in *Proceedings International Test Conference*, IEEE, 1998, pp. 1047–1056.
22. B.H. Seiss, P.M. Trousborst, and M.H. Schulz, "Test Point Insertion for Scan-Based BIST," in *Proceedings European Test Conference*, IEEE, 1991, pp. 253–262.
23. Semiconductor Industry Association (SIA), *International Technology Roadmap for Semiconductors (ITRS)*, 1999 edition.
24. A. Ströle and H.-J. Wunderlich, "TESTCHIP: A Chip for Weighted Random Pattern Generation, Evaluation, and Test Control," *IEEE Journal of Solid State Circuits*, Vol. 26, No. 7, pp. 1056–1063, 1991.
25. N. Tamarapalli and J. Rajski, "Constructive Multi-Phase Test Point Insertion for Scan-Based BIST," in *Proceedings International Test Conference*, IEEE, 1996, pp. 649–658.
26. N.A. Touba and E.J. McCluskey, "Altering a Pseudo-Random Bit Sequence for Scan-Based BIST," in *Proceedings International Test Conference*, IEEE, 1996, pp.167–175.
27. J.A. Waicukauski, E. Lindbloom, E.B. Eichelberger, and O.P. Forlenza, "A Method for Generating Weighted Random Test Patterns," *IBM Journal on Research and Development*, Vol. 33, No. 2, pp. 149–161, March 1989.
28. H.-J. Wunderlich, "Self Test Using Unequiprobable Random Patterns," in *Proceedings 17th International Symposium on Fault-Tolerant Computing*, IEEE, 1987, pp. 258–263.
29. H.-J. Wunderlich and G. Kiefer, "Bit-Flipping BIST," in *Proceedings International Conference on Computer-Aided Design*, IEEE, 1996, pp. 337–343.
30. Y. Zorian, "Testing the Monster Chip," *IEEE Spectrum*, pp. 54–60, July 1999.
31. Y. Zorian, E.J. Marinissen, and S. Dey, "Testing Embedded-Core Based System Chips," in *Proceedings International Test Conference*, IEEE, Washington, D.C, Oct. 1998, pp. 130–143.

Gundolf Kiefer received the Diploma in Electrical Engineering and Computer Science from the University of Siegen, Germany, in 1996. Since September 1996, he is a research assistant at the Institute for Computer Science, University of Stuttgart. His research interests include built-in self-test techniques, design for testability as well as automatic test pattern generation and design automation.

Harald Vranken received an M.Sc. degree in Information Technology in 1992, an MTD (Master of Technological Design) degree in Information and Communication Technology in 1994, and a Ph.D. degree in Electrical Engineering in 1998, all from Eindhoven University of Technology. Since 1998, he is member of the scientific staff at Philips Research Laboratories in Eindhoven, The Netherlands. His research interests include built-in self-test, low-cost high-quality test solutions, and test and debug of VLSI circuits in general. Vranken is author and co-author of over ten journal and conference papers on design-for-testability, system-level test, and application debug.

Erik Jan Marinissen received an M.Sc. degree in Computing Science in 1990 and an MTD (Master of Technological Design) degree in Software Technology in 1992, both from Eindhoven University of Technology. Since 1992, he is member of the scientific staff at Philips Research Laboratories in Eindhoven, The Netherlands. His research interests include all topics in the domain of test and debug of digital VLSI circuits, with an emphasis on core-based testing. Marinissen is author and co-author of over thirty journal and conference papers on design-for-testability and test generation and has several patents pending in these areas. He has presented tutorials on core-based SOC testing at DATE, ETW, ITC, and VTS. Marinissen is member of organizing committees of DATE, ETW, ICCAD, ITSW, LATW, SBCCI, and TECS, and currently serves as Editor-in-Chief of the IEEE P1500 SECT draft standard.

Hans-Joachim Wunderlich received the Dr. rer. nat. (Ph.D.) degree in computer science from the University of Karlsruhe, Germany, in 1986. From 1986 to 1991, he was the head of a research group on automation of circuit design and test at the University of Karlsruhe. From 1991 to 1996 he was a full Professor of Computer Science at the University of Siegen. Since October 1996, he has been the head of the Division for Computer Architecture at the University of Stuttgart, and currently he is the director of the Institute for Computer Science. He is author and coauthor of three books and over 80 papers in the field of test, synthesis and fault tolerance of digital systems. Within the European projects EUROCHIP and EUROPRACTICE he has been a lecturer for courses on VLSI design and test. He is an Editor of the *Journal of Electronic Testing: Theory and Applications*.