Logic BIST With Capture-Per-Clock Hybrid Test Points

Elham Moghaddam, *Member, IEEE*, Nilanjan Mukherjee, *Senior Member, IEEE*, Janusz Rajski, *Fellow, IEEE*, Jędrzej Solecki, *Member, IEEE*, Jerzy Tyszer, *Fellow, IEEE*, and Justyna Zawada, *Member, IEEE*

***Abstract*—Logic built-in self-test (LBIST) is now increasingly used with on-chip test compression as a complementary solution for in-system test, where high quality, low power, low silicon area, and most importantly short test application time are key factors affecting ICs targeted for safety-critical systems. Test points, common in LBIST-ready designs, can help to reduce test time and the overall silicon overhead so that one can get desired test coverage with the minimal number of patterns. Typically, LBIST test points are dysfunctional when enabled in an ATPG-based test compression mode. Similarly, test points used to reduce ATPG pattern counts (PCs) cannot guarantee desired random testability. In this paper, we present a hybrid test point technology designed to reduce deterministic PCs and to improve fault detection likelihood by means of the same minimal set of test points. The hybrid test points are subsequently deployed in a scan-based LBIST scheme addressing stringent test requirements of certain application domains such as the automotive electronics market. These requirements, largely driven by safety standards, are met by significantly reducing test application time while preserving the high fault coverage. The new scheme is a combination of pseudorandom test patterns delivered in a test-per-clock fashion through conventional scan chains and per-cycle-driven hybrid observation test points that capture faulty effects every shift cycle into dedicated scan chains. Their content is gradually shifted into a compactor shared with the remaining chains that deliver responses once a test pattern has been shifted-in. Experimental results obtained for industrial designs confirm feasibility of the new schemes, and they are reported herein.**

***Index Terms*—Design for testability, embedded test, logic built-in self-test (LBIST), scan-based testing, test points.**

I. Introduction

SCAN is one of the most influential and industry-proven structured design for test (DFT) technologies. With a direct access to memory elements of a circuit under test (CUT), scan makes it possible to generate high quality tests and to debug the first silicon, all now supported by EDA tools. Its drawbacks are mainly related to the fact that the vast majority of test time is spent on shifting as virtually all memory elements form shift registers in a test mode. Indeed, 20 000 double-capture test patterns in a design with 400-cell long scan chains would require 8 000 000 and 40 000 shift and capture cycles, respectively. Thus, as low as 0.5% of cycles are spent on testing. The actual test time is also of concern as the scan shift frequency is usually much lower than that of a capture (functional) mode.

Logic built-in self-test (LBIST)—another DFT paradigm—typically employs scan as its operational baseline to achieve, with the aid of some extra on-chip test logic, high quality test while using a limited volume of test data. Classical LBIST applications include different forms of in-field test, detecting infant mortality defects or enabling the use of low-cost and/or low-speed testers that only provide power and clock signals. Interestingly, LBIST keeps up with the demands of new technologies for a valuable test alternative, for example, in the fast-growing automotive electronics market. ICs in this area must adhere to stringent requirements for quality and reliability, which are driven by safety standards such as ISO 26262 and Automotive Safety Integrity Level targets [22]. ISO 26262 compliance requires the adoption of more advanced test solutions. In particular, LBIST should respond to challenges posed by automotive parts and support a number of in-field test requirements: the ability to run periodic tests during functional operations, very short test times due to strict limits on power-up or idle times, high test coverage, or even the ability to incorporate wireless communication into on-chip on-line solutions to ensure highly reliable device operations for the duration of its lifespan.

One of the major problems a conventional LBIST faces is the presence of random-resistant faults. To counteract unacceptably low fault coverage for otherwise feasible pattern counts (PCs), LBIST schemes use either weighted [25], [29], [43], [63], [65] or perturbed [16], [19], [38], [39], [57], [58], [64] pseudorandom patterns. One can also modify a circuit by inserting test points [18] that help to activate faults and observe them by means of control points (CPs) and observation points (OPs), respectively [59]. Several computationally inexpensive techniques were proposed to pick the most suitable locations for CPs and OPs. They are based on fault simulation [7], [24], [55], testability measures [6], [9], [14], [44], [66], or hybrid methods using cost functions [12],

Manuscript received January 23, 2018; revised April 2, 2018; accepted April 18, 2018. Date of publication May 8, 2018; date of current version May 20, 2019. The work of J. Tyszer and J. Zawada was supported by the Polish Ministry of Science and Higher Education under Grant DS-8133/18. This paper was recommended by Associate Editor N. Nicolici. *(Corresponding author: Jerzy Tyszer.)*

E. Moghaddam, N. Mukherjee, J. Rajski, and J. Solecki are with Mentor, A Siemens Business, Wilsonville, OR 97070 USA (e-mail: elham_moghaddam@mentor.com; nilanjan_mukherjee@mentor.com; janusz_rajski@mentor.com; jedrzej_solecki@mentor.com).

J. Tyszer and J. Zawada are with the Faculty of Electronics and Telecommunications, Poznań University of Technology, 60-965 Poznań, Poland (e-mail: jerzy.tyszer@put.poznan.pl; justyna.j.zawada@doctorate.put.poznan.pl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2018.2834441

gradient-based schemes [51], or signal correlation [8]. Since test points need extra gates and flip-flops, a test point population is usually limited by user-defined thresholds.

But it is an LBIST/test compression combined approach that can result in the shortest test time and very high test coverage, on par with the best LBIST and ATPG assets. There are common features that LBIST and test compression may offer. For example, ATPG supports a plug-and-play capability for blocks similar to LBIST reuse with pattern retargeting (using block-level patterns at higher levels). LBIST low-power test remains similar to that of test compression. The same applies to MISR-based diagnosis that has the precision of ATPG but uses LBIST signatures. Finally, for products that need in-field system test, a hybrid approach will always be preferable because of high test quality of ATPG with the autonomous testing of LBIST. Although LBIST test points may reduce pattern counts [36], [52], [67], only ATPG test points, developed specifically for deterministic test, are able to decrease the number of test vectors in a consistent manner [1].

Test compression itself is now coping with the rapid pace of technological changes and the resultant test challenges. With processes scaling down to extremely small feature sizes, using multiple patterning lithography, and moving toward 3-D structures, the number of ATPG test vectors is exploding due to large combinational depths, complex clocking schemes, or excessive tail PCs. Also, in FinFET devices, the gate level abstraction and traditional fault models may not suffice to ensure high-quality and low-DPM metrics for state-of-the-art designs. However, tackling novel timing- and layout-related fault models and patterns based on a post-layout transistor-level netlist (such as cell-aware defects [17]) leads to inflated ATPG test sets that not only require more storage than many testers can afford but also unprecedentedly increase test time.

There are attempts to solve the scan test application time problem. Recall that scan cells are typically controlled by a single scan enable signal, and thus scan chains remain functionally indistinguishable, i.e., they all either shift data in and out or capture test results. A tri-modal scan (TMS) [42] differs from this approach by having scan cells partitioned dynamically to work in three modes where they act as either mission memory elements, sources of test stimuli, or test response compactors. Hence, TMS neither shifts all scan chains nor it captures test responses in all of them. Although stimuli scan chains resemble the conventional scan chains in the shift mode, they do not capture responses and apply test data to the CUT every clock cycle. These are scan chains in the compaction mode that accumulate test responses every clock cycle. The remaining scan cells are kept in the mission mode. Because of test patterns applied every clock cycle, the scheme is time-efficient and allows to complete tests within much shorter durations than those of conventional schemes. Alternatively, it can reach higher defect coverage for otherwise similar test application conditions. TMS has followed several earlier test-per-clock schemes, including built-in logic block observers [31], a circular self-test path [33], and its derivatives [10], [27], [54], E-BIST [4], [53], or certain hybrid techniques [26], [45], [49]. A test-per-clock access to a scan chain can also be accomplished by making scan cells randomly addressable [2], [3].

Certain TMS principles were adapted by the scheme of [68]. It uses shadow registers to capture test results during scan shift. Although the scheme can enhance fortuitous fault detection, it may also inflate the circuit sequential area even above 30% of the original scan cell count.

Although benefits of using LBIST and ATPG in hybrid test schemes are well understood, associated test points remain orthogonal to each other. Therefore, their appearance in hybrid schemes poses nontrivial challenges as they are poorly suited to the other party functionality, they can counteract each other, and they can double related test logic area. In response to these concerns, this paper makes two main contributions. First, we introduce a *hybrid test points* insertion scheme. It minimizes the number of test points needed to enhance performance of both LBIST and test compression mechanisms at the same time.

Next, we propose a scan-based LBIST scheme that aims at achieving a conventional LBIST test quality in much shorter time by applying test patterns every clock cycle, thus increasing the percentage of time used for actual testing. Contrary to TMS and similar schemes, however, our LBIST works with observation hybrid test points that capture faulty effects every shift cycle into flip-flops forming separate scan chains. Their content is continuously shifted into a compactor shared with the remaining scan chains that deliver responses once the entire test pattern has been shifted-in. Since observation points keep the associated flip-flops as a small fraction of the number of scan cells (typically below 2%), the proposed scheme has a low area, is routing friendly, and allows tradeoffs between test time, test coverage, and silicon real-estate.

The remainder of this paper is organized as follows. Sections II and III recall all necessary concepts related to hybrid compression/LBIST architecture and conflict-aware test points, respectively. Section IV is a detailed presentation of the hybrid test points. A test point insertion (TPI) flow is then defined in Section V. Section VI discusses the concept of capture-per-cycle observation test points. The following Section VII presents observation point sites selection procedure. A variety of experimental results obtained for several industrial designs that feature hybrid test points are presented in Section VIII. Section IX concludes this paper. Preliminary versions of this paper were presented at *the IEEE International Test Conference* in 2016 and 2017 [40], [41].

II. Hybrid LBIST/EDT Architecture

Earlier hybrids of LBIST and test compression used separate logic for both schemes despite similarities between their functional objectives. However, sharing certain on-chip resources creates the combination of LBIST and test compression that may perform respectably compared with its traditional hybrid kin. There are several requirements of production and in-system test, though, that a hybrid approach has to address. For instance, a hybrid scheme has to remain a valuable manufacturing test solution with high fault coverage, low test data volume, and a short test time, especially for its LBIST mode running

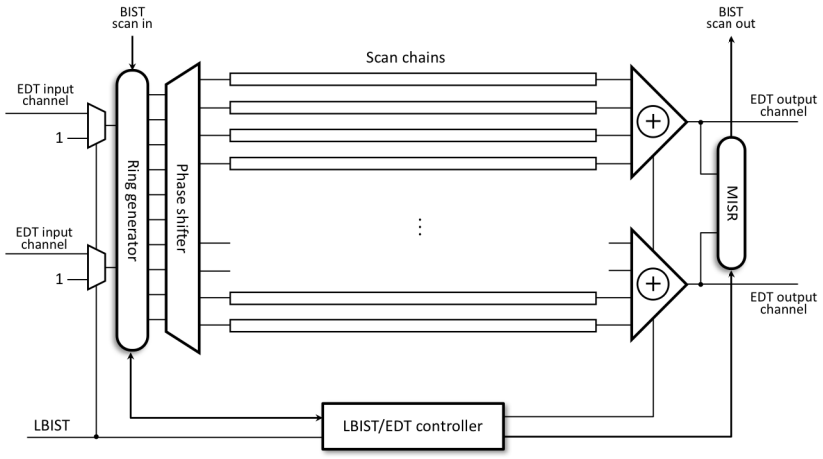

Fig. 1. Hybrid LBIST/EDT architecture.

power-on self-tests. The silicon area, determined by a test controller, a source of test data, and a test response evaluator, should be maintained at the minimum, too.

The above observations in conjunction with the embedded deterministic test (EDT) compression [48] laid the foundation for a hybrid LBIST/EDT scheme shown in Fig. 1. Here, a pseudorandom pattern generator (PRPG), a test response compactor, and the LBIST/EDT controller are shared. Typically, it provides 20%–50% reduction in hardware cost, as indicated by experiments run to compare the total gate count of a hybrid solution versus separate LBIST and EDT schemes where test modules remain disjoint. To reduce external test data and pseudorandom patterns, the hybrid scheme operates in two steps. First, a PRPG targets random testable faults by producing a predetermined number of pseudorandom stimuli. Next, ATPG patterns, applied via an on-chip decompressor, are used to detect random resistant faults. Interestingly, the same hardware—a PRPG—is reused to decompress test cubes. This is only possible in conjunction with sequential test compression. Hence, an n-bit ring generator and a phase shifter make up a sequential decompressor, as shown in Fig. 1. It receives test data through EDT input channels. Two-input multiplexers, placed in the front of the ring generator and controlled by a single enable signal, allow one to feed the ring generator with constant 1s. It facilitates generation of pseudorandom patterns by turning the decompressor into a conventional PRPG.

On the output side, a MISR acts as the LBIST response compactor. The very same unit receives data from XOR trees operating as spatial compactors for scan chain bundles. The same XOR trees produce compressed test data going directly to a tester, if a circuit operates in the test compression mode.

Sharing LBIST logic for the purpose of test decompression was already considered, for example, in [21], [23], [34], [47], and [62]. If the existing LBIST is reused to handle compressed data, then encoding schemes take advantage of low fill rates. Solutions in this class include LFSR coding [30], static [13], [15], [20], [35], [37], [50], [60], [61], and dynamic [5], [11], [48] LFSR reseeding. They are surveyed in [28] and [56].

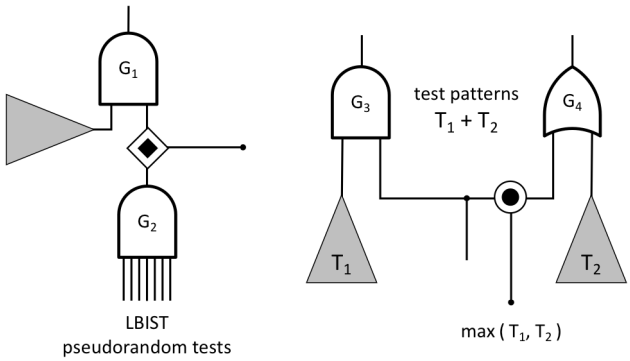

Fig. 2. LBIST and EDT test points.

While sharing basic test components by ATPG and LBIST can be mutually beneficial, orthogonality of their test points may easily elevate a related silicon real-estate beyond acceptable levels. A proper selection of test points and their synergistic usage is therefore of primary interest in hybrid schemes using test points to reduce deterministic test PCs and to boost random testability of a circuit. In the next sections, we comprehensively characterize the TPI process and propose techniques to reduce the area overhead by inserting bifunctional test points at the most appropriate locations.

III. Test Points

LBIST test points are most commonly used to make random resistant logic more testable. For example, to propagate faults through gate G_1 (see Fig. 2), one needs to set the other input of this gate to 1. With pseudorandom patterns driving G_2, the probability of getting 1 on its output is relatively low. This is why having a test point between G_1 and G_2 may resolve this problem.

Consider now a scenario presented in Fig. 2 for gates G_3 and G_4. Let T_1 and T_2 test patterns detect faults propagating to G_3 and G_4, respectively. Clearly, the other input of G_3 must be set to 1. It blocks, however, faults propagating through G_4. Similarly, setting the other input of G_4 to 0 stops propagation of faults via G_3. Because of these mutually opposed

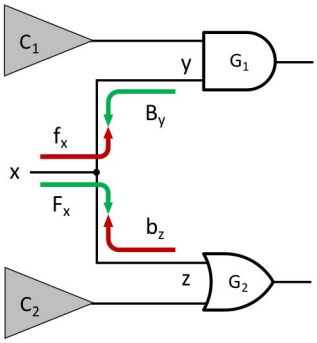

Fig. 3. Conflict on internal lines.

requirements for G_3 and G_4 driven by the same stem, faults in both groups cannot be detected by the same test patterns, and the resultant PC is, therefore, equal to $T_1 + T_2$. Simultaneous detection of these faults would be possible by placing a CP on one of the stem branches to resolve the conflict. For instance, a test point on the left branch achieves 1-controllability of this line, whereas a test point on the right branch allows 0-controllability of this net. As a result, the number of test patterns becomes max $\{T_1, T_2\}$. Interestingly, if $T_1 \approx T_2$, then the PC can be approximately halved. Hence, test points placed in this type of locations might significantly reduce the overall pattern count.

Contrary to LBIST test points, conflict-aware (or EDT) test points [1], [32] recalled above aim primarily at reducing ATPG PCs and test data volume. A key feature of the EDT TPI scheme is its ability to identify and resolve conflicts between ATPG-produced signals. As a result, it increases the number of faults detected by a single pattern, and reduces both the number of deterministic tests and test data volume, leading eventually to shorter ATPG and test times.

Procedures used to place EDT test points are comprehensively presented in [1]. Here, we just recall key concepts. Consider a circuit of Fig. 3. Let cones of logic C_1 and C_2 host D_1 and D_2 faults, respectively. Since the only propagation paths from C_1 and C_2 pass through gates G_1 and G_2, the conflict-aware TPI will assess the degree of inconsistency between signals needed to propagate faults (backward justifications) and the resultant forward implied values on stems. For example, to propagate D_1 faults through G_1, input y must be set to 1 at least $B_y = D_1$ times. Similarly, we will get the number b_z of 0s enabling fault propagation from C_2. Clearly, $b_z = D_2$. Metrics b and B can also be regarded as the number of blocked faults, if a given line was set to 1 and 0, respectively [1].

The values of b and B help to estimate the numbers f and F of forward-implied 0s and 1s, due to backward justifications. Back to Fig. 3, $f_y = b_z + f_x$, i.e., 0s at y are implied by 0s on branch z plus 0s on stem x (as a result of earlier computations). Next, given a gate and its input f/F values, simple formulas [1] are used to determine the output values of f/F. Thus, one can assess conflicts occurring at a given fan-out branch by comparing the associated values of B and f (or b and F – see branches y and z in Fig. 3). It is the result of this comparison that determines the type and the location of a control test point resolving ATPG-induced assignment conflicts.

TABLE I
EDT TPs VERSUS LBIST TEST COVERAGE

Gates	Scan cells	Baseline [%]	EDT TPs	LBIST TC [%]
1.5M	75K	79.31	2K	84.08
2.6M	154K	76.46	1.5K	83.91
3.6M	41K	81.43	1.1K	84.63
2.6M	150K	83.29	3K	85.79

TABLE II
LBIST TPs VERSUS ATPG PC

Gates	Scan cells	Baseline	LBIST TPs	ATPG PC
2.1M	148K	10,175	3K	6,335
2.2M	143K	23,089	2.3K	**23,777**
4.4M	308K	69,606	4.5K	48,905
1.2M	63K	8,539	1.2K	**10,759**

IV. HYBRID TEST POINTS

Having two complementary classes of test points, the question now is if EDT test points can improve LBIST-based test coverage, or whether LBIST test points can reduce ATPG-based PCs. Industrial experimental results addressing these concerns are reported in Tables I and II. The first two columns of both tables list the number of gates and scan cells. The third column of Table I reports the baseline test coverage for 10K pseudorandom patterns, while the third column of Table II provides the baseline ATPG PC. Now, given the number of EDT test points (EDT TPs in Table I), the last column provides the resultant LBIST test coverage. Similarly, the last column of Table II reports the number of test patterns needed to yield test coverage of the baseline case with LBIST test points inserted (column LBIST TPs). In all experiments the number of test points varies between 1% and 2.7% of memory elements deployed in the designs.

As presented in Table I, EDT test points may fairly improve designs' random testability, while resolving internal conflicts, though they do not guarantee complete fault coverage yet. On the contrary, LBIST test points [51] cause less predictable outcomes. Table II reveals that they may affect test PCs either way (cases of undesired PC grow are highlighted). Neither EDT nor LBIST test points are, therefore, exclusively fitted for hybrid test solutions. In fact, having both types of test points on chip may compromise test data compression, random testability, and a silicon area. Hence, we propose an approach based on insertion of *hybrid test points*.

A. Fault Propagation

The number of faults propagating through a net is one of the key factors used to identify the most suitable locations of hybrid test points. A fault propagation analysis follows a circuit's gate-level order. Starting from the first level, the number of faults occurring at the output of each gate is computed as a sum of faults reaching its inputs. A fan-out distributes faults to its branches in a way that corresponds to their COP-based [6] observabilities. Consider a circuit in Fig. 4. Let D_i

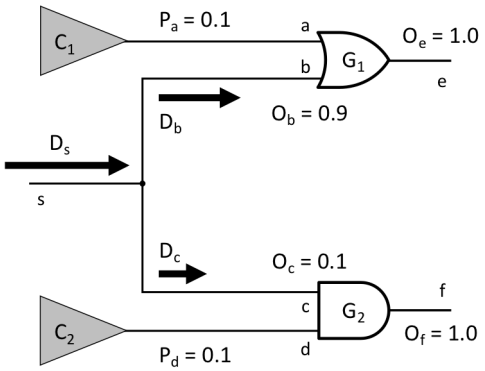

Fig. 4. Fault propagation.

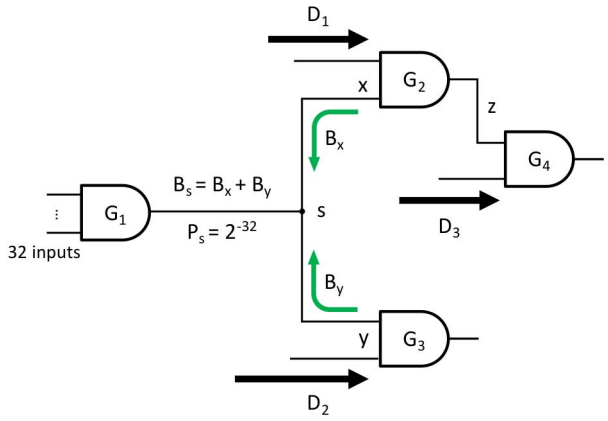

Fig. 5. Hybrid conflict.

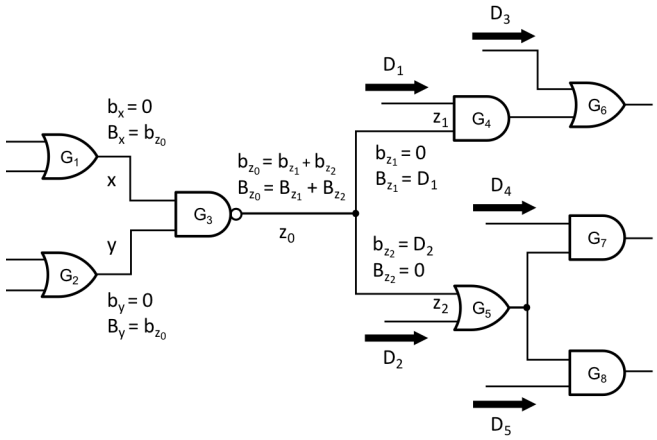

Fig. 6. Computation of metrics b and B.

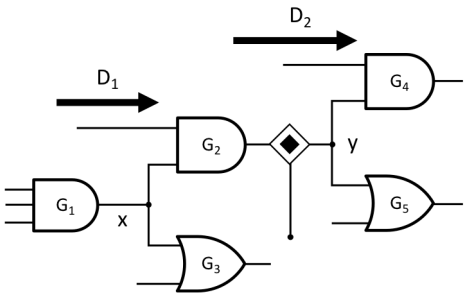

Fig. 7. Example of hybrid CP.

be the number of faults propagating via line i (also referred to as faults D_i and represented by proportionally sized arrows in Figs. 4–7). Also, let P_i be 1-controllability of line i, i.e., a difficulty of setting line i to 1, while O_i be line i observability, which is the probability of enabling a fault propagation path from that particular line to an output. Let us assume that $P_a = P_d = 0.1$. Since primary outputs e and f are fully observable, i.e., $O_e = O_f = 1.0$, we have $O_b = O_e \cdot (1–P_a) = 0.9$ and $O_c = O_f$. $P_d = 0.1$. Clearly, line b is much better observable than c, thus there are more faults propagating through b than those reaching c. In general, the number of faults propagating from stem s through branch x is given by

$$D_x = \alpha \cdot D_s \tag{1}$$

where α is a line x observability divided by the sum of observabilities associated with all stem's branches.

B. Control Points

Consider now the circuit of Fig. 5. From a perspective of EDT test points, line x has to be set to 1 at least $B_x = D_1 + D_3$ times (see Section III). Clearly, setting x to 0 blocks propagation of faults D_1, and then faults D_3. On the other hand, to facilitate propagation of faults D_2, we need to assert branch y at least $B_y = D_2$ times. In the worst case, therefore, to propagate faults D_1, D_2, and D_3, stem s must be set to 1 at least $D_1 + D_2 + D_3$ times, i.e., $B_s = B_x + B_y$. To estimate the degree of conflict it may cause (that could be resolved by EDT test points), we need to know values of f and F at stem s that depend on its driving network. Evidently, having a conflict here may inflate an ATPG PC. Furthermore, 1 showing up at the output of gate G_1 is a very unlikely (1 out of 2^{32}) random event because of its 32 inputs. To resolve this conflict, the LBIST test points insertion [51] would identify the output of G_1 as a suitable candidate for the OR type control test point allowing to increase the probability of faults D_1, D_2, and D_3 moving forward and possibly getting detected.

Attractive as the output of gate G_1 seems for EDT and LBIST test points, a careful assistance is needed in identification of such mutually beneficial locations. This is a *hybrid conflict* that allows us to determine nets whose high demands for signals of a given polarity face low probabilities of getting them at those sites. To identify a hybrid conflict, we assign EDT metrics b and B to each relevant line of a design. Each fan-out stem is assigned, in a levelized-order, the logic values of 0 or 1. These values propagate then forward until they hit input(s) of a gate as its noncontrolling value (this propagation can continue provided all inputs of the gate are set to the noncontrolling value). If the forward propagation implies values on other stems and their branches, then these nets will not be evaluated separately. Since the controlling value at the input of a gate blocks fault propagation through that gate, one can determine metrics b and B of how many times the noncontrolling values have to be assigned to this input to make the gate output sensitive to the incoming faults. Also, b/B values accumulate, taking into account a gate's inversion, the corresponding values of b/B obtained earlier for the gate output.

By going backwards, a starting fan-out stem x with n fan-out branches receives eventually

$$b_x = b_{x1} + b_{x2} + \cdots + b_{xn} \quad (2)$$

$$B_x = B_{x1} + B_{x2} + \cdots + B_{xn}. \quad (3)$$

The stem values are now justified in the process of back tracing, within a fan-out free region (FFR) it is driven by, thus producing values of b and B for the FFR relevant nets. Let the gate G_1 output have $b_s = 0$ and $B_s = 100$ (Fig. 5). Since 1 is a noncontrolling value for the AND gate, each input k of G_1 gets $B_k = 100$, and the process continues backwards. On the other hand, we cannot justify b_s on the inputs of the same gate, as 0 is the controlling value here.

Example: Let us determine the values of b and B in a circuit of Fig. 6. The number of 0s required on stem z_0 to enable propagation of faults through all relevant gates matches the number of blocked faults when z_0 is set to 1. As can be verified, after injecting 1 at z_0, the forward value propagation continues until gates G_4, G_7, and G_8, where faults D_1, D_4, and D_5 remain observable due to 1 reaching these gates. Now, going backward, 1s at branch z_2 block faults D_2. This implies that branch z_2 must be set to 0 at least D_2 times, i.e., $b_{z2} = D_2$. Moreover, $b_{z1} = 0$ as 0 is a dominating value for gate G_4. As a result, $b_{z0} = b_{z1} + b_{z2} = D_2$, see (2). To produce 0 at the output of G_3, all its inputs must be asserted. Thus, the numbers B_x and B_y of 1s required at x and y are equal to $B_x = B_y = b_{z0}$. The backward justification of B stops at G_1 and G_2 as 1 is a dominating value for the OR gate. Similarly, the number of 1s required on stem z_0 is obtained by setting z_0 to 0 and propagating it forward. It passes G_4 and reaches G_6 as a nondominating value. Going further backward, we get that the number B_{z1} of 1s required to propagate faults D_1 becomes equal to D_1. From (3), $B_{z0} = B_{z1} + B_{z2} = D_1 + 0$. Since 1 at the output of NAND gate G_3 can be produced in different ways, the backward justification of B stops here.

Besides metrics b and B, the hybrid conflict analysis employs COP-based controllability measures. Let p_x and P_x be the probabilities of getting values of 0 and 1 at line x, respectively. We propose to measure the degree of conflict between a logic value enabling fault propagation and the probability of its occurrence at the same location as follows:

$$c_x = -b_x \log_{10} p_x \quad (4)$$

$$C_x = -B_x \log_{10} P_x. \quad (5)$$

Since $p_x \leq 1$ and $P_x \leq 1$, the larger the values of c_x or C_x, the smaller the chance that desired fault propagation conditions can be matched by randomly produced signals. For example, if $b_x = 15$, and $p_x = 0.0001$, then $c_x = 60$, which contrasts $c_x = 4.52$ obtained for $b_x = 15$ and $p_x = 0.5$. The former case indicates that a test point at x is highly desired and should alleviate the backward justification problem, in addition to enhanced random controllability of x. Note that a control test point type depends on which metric determines the dominating type of conflict. The AND CP is used if $c_x > C_x$, and the OR CP is inserted otherwise.

A new CP affects metrics b, B, as well as signal controllabilities at several locations and rearranges the existing conflicts. Consider the circuit in Fig. 7. Recall that the number B_x of 1s needed at stem x (or the number of blocked faults when line x is set to 0) is equal to $D_1 + D_2$. If the OR CP is added at the output of G_2, then B_x reduces to D_1, as this CP can stop propagation of 0s. Also, the probabilities of having 0 and 1 at stem y change as a result of the same CP insertion. Subsequently, these COP metrics propagate forward, hence also changing conflicts in surroundings of G_1. In order to update affected metrics in the most efficient way, the following steps are performed, both starting from a newly inserted control point.

1) Move backward and recompute b and B as long as their former values and the new ones differ. At the same time, update conflicts (4) and (5) at all visited nets.
2) The signal probabilities are recalculated until the difference between the previous values and the new probabilities becomes smaller than a certain threshold. Conflicts (4) and (5) are updated as well.

C. Observation Points

In many circuits, there are faults difficult to detect due to severe propagation conditions. In those cases, observation points can shorten propagation paths. To identify low observability lines propagating, at the same time, prominent number of faults, we compute *observation performance* Ω_x for line x as follows:

$$\Omega_x = -D_x \cdot \log_{10} O_x \quad (6)$$

where D_x and O_x are the number of faults propagating through net x, and this line observability, respectively. Since $O_x \leq 1$, the larger the value of D_x, the more efficient observation point x can make. For example, if $D_x = 1000$ and $O_x = 10^{-4}$, then $\Omega_x = 4000$. However, the small number of faults, say $D_x = 10$, causes the observation performance to drop to $\Omega_x = 40$. The former case indicates a desired observation point at x, as it may facilitate detection of a significant number of faults.

Since a newly inserted observation point affects observabilities of other nets and changes fault propagation paths, we revise both the remaining testability metrics and faults propagating within an area affected by the observation point as follows.

1) Starting from the new observation point, move backward toward inputs and update observability of each traversed line marking all visited stems until no further changes can be made.
2) Given all visited stems, pick the lowest-level stem s.
3) Beginning with stem s and by using (1), move toward outputs in a gate-level order revising fault propagation data until no further changes can be made.

V. Test Point Insertion

The following greedy steps are carried out to identify and insert hybrid test points. A circuit is processed in a gate-level order in two major steps. First, we compute COP-based controllabilities and observabilities for each net. Next, starting from the primary inputs, faults are propagated forward, and subsequently performance of a candidate observation point is

determined for each net using (6). The last iteration of the first step computes metrics b and B for relevant lines by processing stems in a levelized-order. Initially all these metrics are set to 0. Since b_s and B_s are equal to the number of blocked faults when stem s is set to 1 and 0, respectively, this contrapositive rule allows a circuit tracing to obtain estimations of b_s and B_s, and subsequently logic values assigned to nets within an FFR driving stem s. At the same time, the hybrid conflicts for CPs are computed according to (4) and (5). As a result, the first part of this procedure yields two sorted in a descending order lists representing the best locations for control and observation points.

The number of test points and its breakdown into control and observation points is typically a design-dependent factor. To guide TPI, we propose to monitor the resultant test coverage, as we keep adding new control and observation points. Let T_x be fault x detection probability

$$T_{x/0} = P_x \cdot O_x \tag{7}$$

$$T_{x/1} = p_x \cdot O_x \tag{8}$$

where $x/0$ and $x/1$ are line x stuck-at-0 and stuck-at-1 faults. The values of p_x and P_x represent the degree of difficulty in setting line x to 0 and 1, respectively. Test coverage for n test patterns can be then estimated as the average over the entire list of testable faults by using the probabilities of detecting faults by at least one out of n test patterns.

In the second phase, we repeatedly remove the top of the CP list, insert the corresponding CP, and determine its impact on test coverage. Next, we withdraw the newly inserted CP, restore the circuit previous state and perform the same actions again, but now we consider an observation point from the current top of observation point list. Which test point should be added into the design is decided based on the resultant test coverage numbers obtained in each case. The rejected test point returns to its list. Note that after inserting every test point, all testability measures and other metrics are updated. For a CP, this process is run as described in Section IV-B. After inserting an observation point, observabilities of internal nets and fault propagation paths are recomputed as presented in Section IV-C. The TPI procedure iterates until either the number of inserted test points matches the desired and user-defined number of test points that can be added into the design or a target test coverage is reached.

VI. Capture-Per-Cycle Test Points

In addition to enriching hybrid test solutions, the test points presented in the previous sections can play a crucial role in a novel LBIST scheme that we will discuss in the next two sections. In this approach, observation hybrid test points capture faulty effects every shift cycle into dedicated flip-flops that form separate scan chains. Their content is gradually shifted into a compactor shared with the remaining chains that deliver responses once the entire test pattern has been shifted-in. Furthermore, hybrid CPs facilitate propagation of faults toward scan chains working with observation points. Consequently, the scheme either significantly reduces test application time while preserving high fault coverage or allows applying a much larger number of vectors within the same time interval.

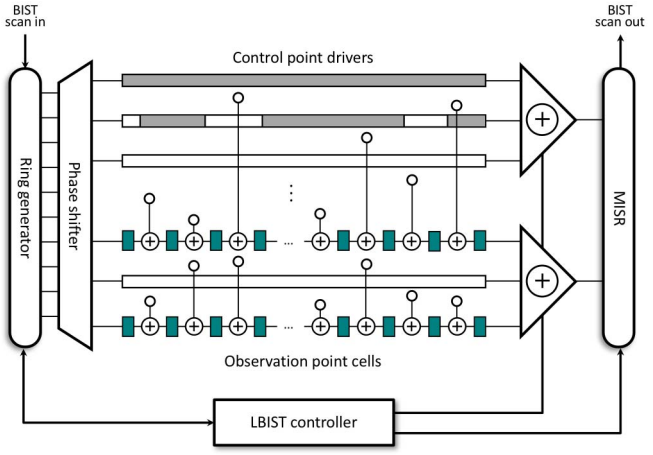

Fig. 8. New LBIST architecture.

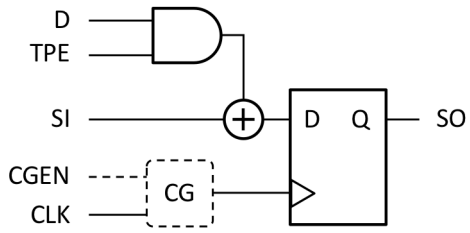

Fig. 9. Scan cell for observation point.

A basic architecture of the proposed LBIST scheme appears in Fig. 8. The vast majority of scan cells form conventional scan chains (white colored), i.e., they operate either in the shift (after asserting the scan enable signal) or in the capture mode. Since control test points set internal lines to specific logic values, all flip-flops associated with them work only in the shift mode and can be arranged in two different ways: either dedicated scan chains host only drivers of CPs (such as the top chain in Fig. 8) or scan cells driving CPs are interspersed among other scan cells (the second chain in the figure).

The green-colored cells serving the observation points are arranged into independent scan chains that accumulate test responses using XOR gates placed in the front of scan cells (Fig. 9). It allows one to encapsulate shift and capture functionality within a single clock cycle. The global test point enable signal activates observation points in the test mode. Test results received from CUT through inputs D are then XOR-ed with data provided by adjacent scan cells. A clock gating may also be used to actually enable or disable compaction scan chains after asserting the corresponding control (CGEN) signal. Although compaction chains may not be fed by a phase shifter, this connectivity enables uniform scan-integrity tests.

A typical LBIST session employing the proposed test architecture can be described as follows. Initially, the first pseudorandom test pattern is shifted-in, and all scan chains capture a test response. Next, when the response is shifted-out with the subsequent test pattern filling gradually the regular scan chains and the CP drivers, the content of these scan cells

TABLE III
DESIGN CHARACTERISTICS

	Gates	Scan cells	Scan chains	EDT In ; Out	Chain / channels	TPs
D1	218K	14K	20 x 710	1 ; 1	20x	568
D2	1.2M	72K	400 x 181	4 ; 4	100x	1,500
D3	2.1M	143K	400 x 359	4 ; 4	100x	3,000
D4	3.3M	326K	400 x 814	4 ; 4	100x	6,500
D5	1.5M	75K	280 x 268	2 ; 2	140x	1,500
D6	2.6M	160K	400 x 401	4 ; 4	100x	3,200
D7	1M	57K	400 x 144	4 ; 4	100x	1,200

TABLE IV
ATPG PC

	PC	TC [%]	EDT	Hybrid
D1	10,937	99.99	6,522	4,408
D2	8,632	96.99	5,184	5,406
D3	24,531	99.55	8,286	7,701
D4	3,518	96.48	2,205	1,922
D5	15,637	97.19	13,052	12,485
D6	4,796	98.97	3,291	2,677
D7	15,467	91.46	5,248	3,712

TABLE V
LBIST TEST COVERAGE

	TC [%]	LBIST [%]	Hybrid [%]
D1	40.89	41.08	68.59
D2	78.53	85.72	90.41
D3	80.74	85.88	89.57
D4	84.38	89.41	92.55
D5	79.35	81.39	87.58
D6	90.34	95.41	97.07
D7	73.37	80.84	87.54

becomes stimuli feeding the circuit every clock cycle. It allows the observation-point-scan-chains to capture and accumulate test responses every clock cycle. Furthermore, the resultant response enters a MISR in a single-bit-per-cycle regime. Note that we restrict the input variables on which CUT depends in such a way that test responses captured by scan chains operating in the compaction mode do not drive the design. It greatly simplifies fault simulation. Clearly, when the regular scan chains are fully loaded with a new test pattern, these scan chains as well as the compaction chains capture a test response (the regular chains in a conventional manner). As a result, the circuit is tested in accordance with the test-per-clock paradigm, while preserving all benefits of the test-per-shift approach.

VII. TEST POINT SITES

Since the proposed LBIST scheme deploys scan chains working either in a test-per-pattern or a test-per-clock fashion, these two groups of scan chains capture test responses within different time intervals. Therefore, to identify the most suitable test point locations, the analysis of hybrid control and observation points presented in Section IV must account for these new operational principles.

Typically, testability measures grant all outputs (and pseudo-outputs, e.g., regular scan cells) full observability all the time. However, in our scheme the observability metrics are adjusted to observation test points that capture faulty effects every shift cycle. Hence, we modify observability metrics so that every single observation point x gets $O_x = 1.0$ during all shift and capture cycles, whereas observabilities for the remaining scan cells (that collect test responses only during capture cycles at the end of the scan shift-in phase) are set to $1/n$, where n is the size of the longest scan chain. Having modified testability measures, the TPI procedure of Section V can be carried out.

It is also worth noting that observabilities of regular scan cells (set to $1/n$) discourage the TPI procedure from assuming that conventional scan chains may suffice as observation points. Instead, TPI tries to identify bottlenecks in a design, where fault propagation is systematically hampered and can be cured by adding observation points of the new class.

VIII. EXPERIMENTAL RESULTS

The objective of the experiments reported in this section was to assess effectiveness of the hybrid test points applied within two different frameworks. First, we demonstrate how test points enhance both test compression and random testability in hybrid EDT/LBIST setup. Next, we show that the proposed test points can either yield desired test coverage faster than traditional LBIST schemes, or they can offer visibly higher coverage numbers in much shorter time than its conventional counterparts. All experiments reported in this paper have been conducted for testable static faults on large and complex industrial circuits representing different design styles and scan methodologies. In order to reduce the impact on timing closure, we have restricted the number of control test points on a single path to five in a manner similar to that of a technique presented in [1].

A. Hybrid EDT/LBIST

The first group of experiments used designs listed in Table III. It includes the number of gates, the number of scan cells, scan architecture, the EDT interface, and the target input compression. The last column provides the number of test points employed. This number is a 2% fraction of the entire scan cells population but design D1 where it is equal to 4%.

To start with, we examine how different types of test points affect both the ATPG PC and LBIST test coverage. For each test case, we used: 1) exclusively EDT test points [1]; 2) LBIST test points resting on traditional COP-based testability measures [51]; and 3) the hybrid test points. All configurations assume the number of test points shown in Table III.

Table IV presents ATPG PCs. Column PC lists the baseline (no test points) PC with the corresponding test coverage in column TC. The next columns report the number of patterns necessary to get the baseline test coverage for different TPI configurations. Note that in all test cases but D2, the hybrid

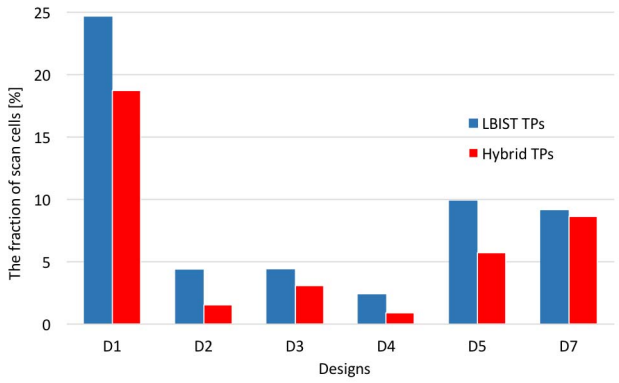

Fig. 10. Number of test points needed to reach 90% test coverage.

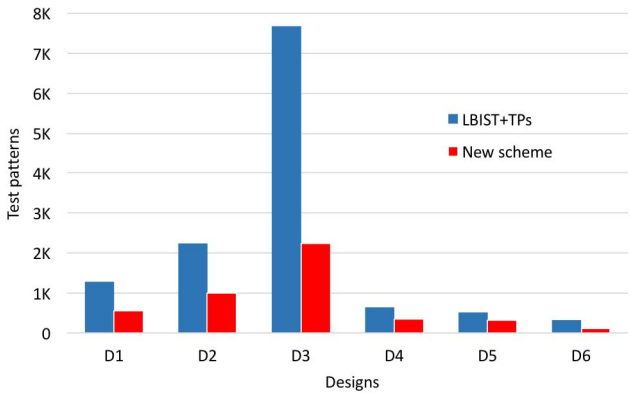

Fig. 11. Patterns needed to reach 90% test coverage.

TABLE VI
DESIGN CHARACTERISTICS

	Gates	Scan cells	Scan chains	Longest chain	TC [%]	Faults
D1	1.2M	85K	427	200	86.54	3.4M
D2	901K	36K	359	100	85.78	2.1M
D3	1.2M	72K	382	190	78.53	3.8M
D4	453K	45K	226	200	83.54	1.3M
D5	2.6M	160K	1,015	158	90.43	7.9M
D6	1.6M	144K	700	207	88.83	3.2M

test points result in either lower PCs or similar to those when only EDT test points are used. These test points resolve several ATPG-induced conflicts, and yet some conflicts can only be fixed by EDT test points. Nevertheless, as can be seen from the table, the hybrid test points provide, on the average, a 2.3× PC reduction.

Table V reports test coverage for the same hybrid test points as above but used with LBIST applying 10K pseudorandom test patterns. Column TC is the baseline test case. Once again, the hybrid test points yield higher test coverage than that of conventional LBIST test points (column LBIST). Even poor random testability of design D1 is visibly improved after inserting the hybrid test points.

It might be of interest to see how many test points are needed to achieve a given target LBIST test coverage. Fig. 10 shows such results for a 90% threshold. The blue bars represent LBIST test points, whereas the red bars correspond to hybrid test points. All numbers are reported as a fraction of the total number of scan cells deployed in the designs. D6 is not present as its baseline test coverage is already above 90% limit (see Table V). The proposed TPI scheme requires, on the average, 1.8 times fewer test points than the conventional testability-based TPI methods. Clearly, this result is indispensable for silicon real estate savings.

B. Capture-Per-Cycle Hybrid Test Points

The second group of experiments is to examine performance of the LBIST scheme of Section VII. This is done by comparing its test coverage with test coverage provided by a conventional LBIST with hybrid test points. Experiments were performed on six industrial designs that are currently in high-volume production. Information about these designs, including the number of gates, the number of scan cells, the size of the longest scan chain, a baseline (no test points) test coverage after applying 10K pseudorandom test patterns, and the number of testable faults is reported in Table VI.

Experimental results presented in the following part of this section were obtained for single-capture stuck-at patterns, i.e., a clock pulse is only applied once between scan-in and scan-out operations to capture test results. In preparations for the experiments, we generated two groups of hybrid test points for each design. The first group is intended for circuits with the conventional LBIST test points, further referred to as a *reference* case, where all regular scan cells and extra observation points are fully observable once a test pattern is entirely shifted into scan chains, and a circuit becomes ready to capture a test response. Test points are inserted by employing the algorithm of Section V. We apply 10K pseudorandom test patterns in this case.

Test points of the second group are used in conjunction with the new LBIST framework. These test points increase the detection probability of random-resistant faults and enable detection of faults during scan-shift cycles. In particular, observation points allow faulty effects to be captured every shift cycle, while CPs facilitate fault propagation toward additional scan chains operating in the compaction mode. To find the best test point locations, we employ the TPI algorithm of Section V with modified line observabilities of Section VII. As shown earlier, this is accomplished by assigning regular scan cells the observability value of $1/n$ (n is the size of the longest scan chain), whereas the remaining observation sites have their observabilities set to 1.0. Again, the target PC comprises 10K *base* pseudorandom patterns. However, intermediate test patterns are also taken into account. Let T_i and T_{i+1} be two subsequent base test patterns. In this case, intermediate patterns consist of capture values corresponding to T_i and PRPG-produced values that correspond to T_{i+1}. Clearly, the first intermediate vector of a given bundle is equal to a test response obtained for T_i, while the remaining ones are generated on the fly as a result of scanning-out a test response of T_i and scanning-in a new pattern T_{i+1} at the same time. This process repeats until R pseudorandom base test patterns

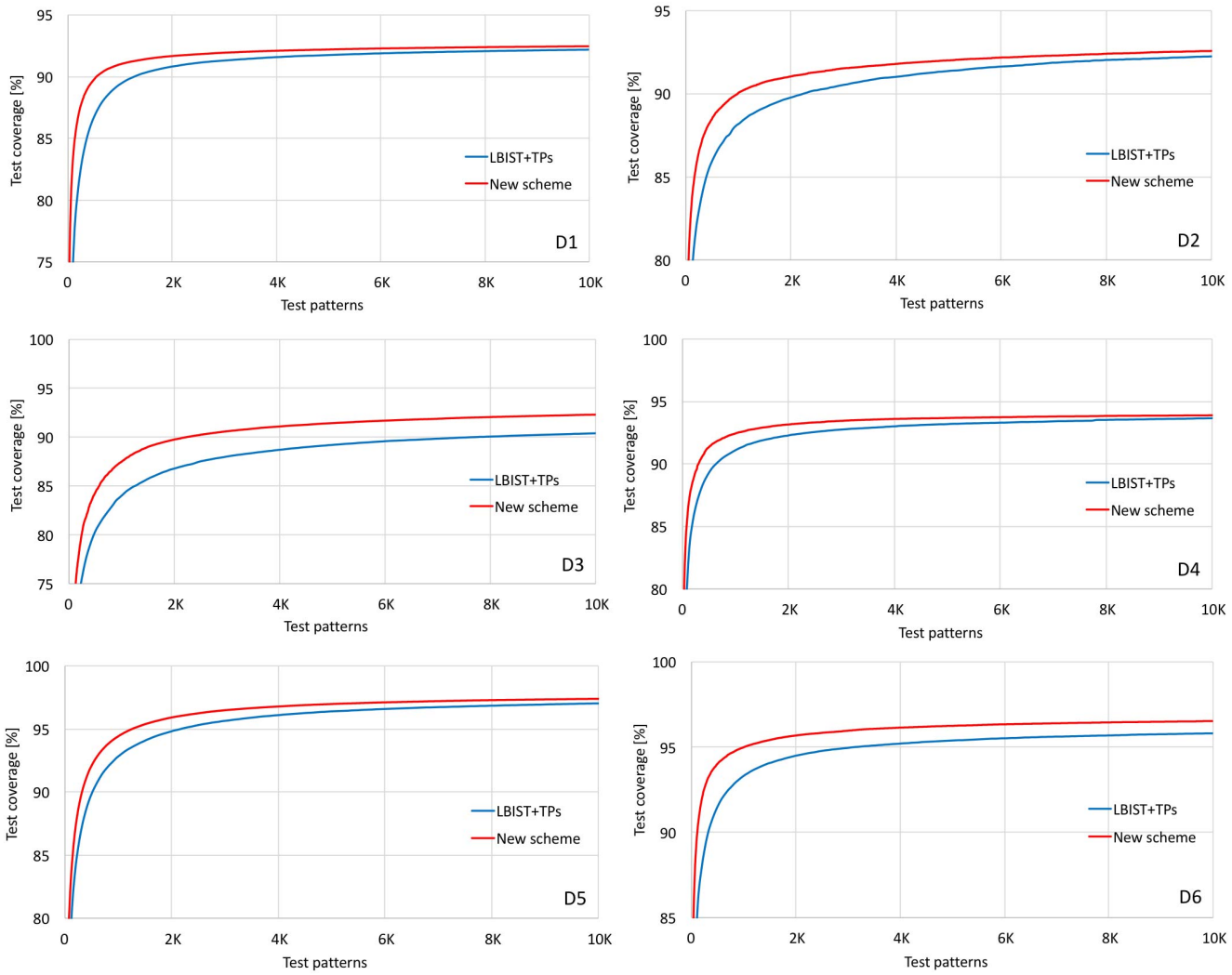

Fig. 12. Test coverage as a function of PC.

TABLE VII
EXPERIMENTAL RESULTS FOR STUCK-AT FAULTS

	CPs	OPs	1,000 patterns		2,000 patterns		4,000 patterns		10,000 patterns	
			TC [%]	ΔTC [%]	TC [%]	ΔTC [%]	TC [%]	ΔTC [%]	TC [%]	ΔTC [%]
D1	1,180	520	89.46	1.58	90.85	0.84	91.59	0.51	92.19	0.28
D2	380	330	88.23	1.87	89.52	1.57	91.03	0.77	92.25	0.33
D3	420	1,080	84.13	3.46	86.86	2.94	88.72	2.38	90.39	1.92
D4	180	720	91.16	1.33	92.33	0.85	93.02	0.59	93.67	0.23
D5	900	2,300	92.91	1.58	94.88	1.07	96.11	0.68	97.03	0.37
D6	1,000	1,800	93.37	1.66	94.53	1.17	95.22	0.93	95.81	0.72

are generated, where R is a user-defined quantity. The total number P of patterns applied to the CUT by the new LBIST is given by

$$P = R + (R - 1) \cdot n \tag{9}$$

where $(R - 1) \cdot n$ is the number of intermediate test patterns obtained when the regular scan chains are shifting, with n, as earlier, being the size of the longest scan chain.

The experimental results obtained for stuck-at faults are listed in Table VII. The first two columns report the breakdown of test points (CPs and OPs) obtained by deploying the algorithm of Section V. The very same number of control and observation points are used for every new LBIST test case. The number of test points is 2% of the entire scan cell population. The next columns of Table VII give test coverage TC obtained in the reference case, and then test coverage increase

TABLE VIII
EXPERIMENTAL RESULTS FOR STUCK-AT FAULTS

	1,000 patterns						10,000 patterns					
	2 captures		3 captures		4 captures		2 captures		3 captures		4 captures	
	TC [%]	Δ TC [%]	TC [%]	Δ TC [%]	TC [%]	Δ TC [%]	TC [%]	ΔTC [%]	TC [%]	Δ TC [%]	TC [%]	Δ TC [%]
D1	88.72	3.17	85.97	6.06	85.9	5.46	93.28	0.51	92.31	2.18	92.31	2.25
D2	87.08	2.35	85.59	2.75	84.35	3.22	91.56	0.62	90.48	1.05	89.36	1.47
D3	87.99	2.46	88.98	2.19	89.26	2.48	92.59	1.4	93.18	1.26	93.61	1.16
D4	91.22	1.51	90.99	1.65	90.61	1.98	93.59	0.46	93.37	0.61	93.26	0.61
D5	91.22	1.9	88.86	2.11	86.82	2.43	96.75	0.66	96	0.99	95.06	1.06
D6	92.75	1.77	91.07	2.57	89.65	2.99	95.39	0.77	94.45	1.26	93.82	1.48

ΔTC is reported for the new LBIST scheme after applying 1K, 2K, 4K, and 10K base pseudorandom patterns. Consider, for example, 2K base patterns. From (9) we get that the new method uses then $2000 + 1999\ n$ test vectors. If $n = 100$, then the total number of actually applied patterns amounts to almost 202K.

Fig. 11 illustrates how many test patterns it takes for both the conventional LBIST test points and the proposed scheme to reach target 90% test coverage—an industry-wide accepted automotive electronics in-field-test-quality standard. As can be seen, superiority of the new solution is clearly pronounced. The new LBIST outperforms the traditional approach, on the average, by a factor of 2.4, i.e., it is capable of providing 90% test coverage more than 2.4 times faster than its state-of-the-art conventional counterpart. While the transition to fully computerized vehicles is speeding-up and breaking into the mainstream automobile market, the above result is definitely worth of appreciation.

Fig. 12 provides a much more comprehensive illustration of test coverage as a function of test time. As can be seen, in all test cases, the proposed method, represented by the red curve, reaches a given test coverage level in much shorter time than the conventional LBIST with test points (the blue curve) does. Furthermore, test coverage provided by the new scheme for designs D3 and D6 remains noticeably higher (after 10K base patterns) than that of the reference case.

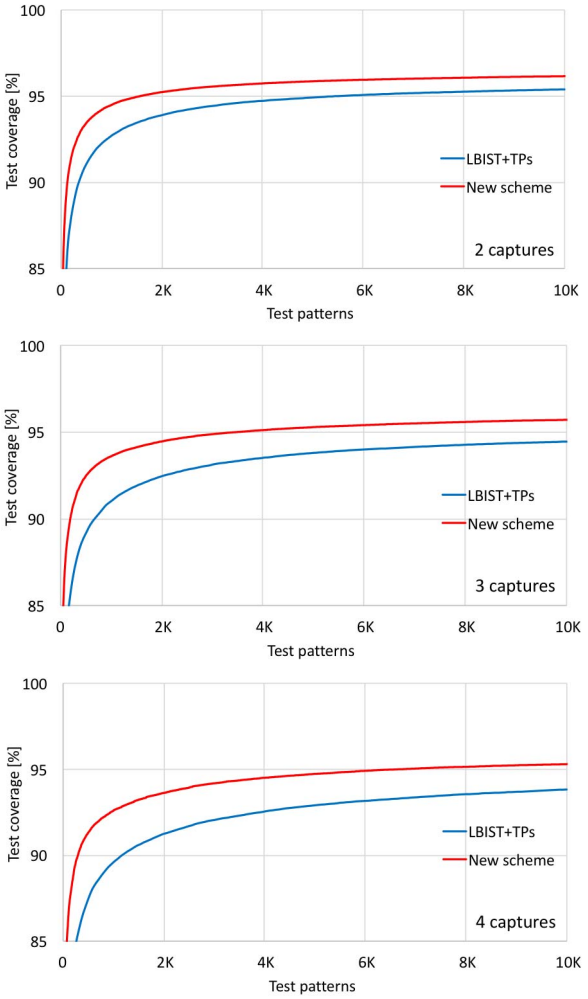

Fig. 13. Test coverage as a function of PC—design D6.

C. Multicycle Test

The results presented in the previous sections were obtained for single-capture test patterns. Since test coverage can often be increased by incorporating multiple capture cycles, in so called multicycle tests a test pattern being applied to a combinational logic is the output response of a circuit to the previous test pattern. The main drawback of a multicycle test is that we observe a test response only at the last capture cycle. Consequently, an error showing up in the middle of the capture mode may not be caught by the last capture clock pulse.

To increase the number of faults detected by multicycle tests, we propose to employ the experimental setup of Section VIII-B with hybrid observation points being active in the capture mode. Note that observation points record now faulty effects every shift and capture cycle, thus increasing the number of faults detected by a single pattern. To examine effectiveness of this scheme, we use again the industrial designs of Table VI with the breakdown of test points shown in Table VII. Similarly as in Section VIII-B, Table VIII reports a test coverage increase (ΔTC) over conventional LBIST with hybrid test points after applying 1K and 10K base pseudorandom patterns. Results are presented for 2, 3, and 4 capture cycles. Clearly, there is a significant test coverage improvement in all test cases.

Consider now design D6. Given the number of capture clock cycles, Fig. 13 illustrates how test coverage increases with the increasing number of test patterns. The blue curve represents conventional LBIST with hybrid test points, whereas the red one corresponds to the new scheme observing faulty effects every shift and capture cycles. The superiority of the proposed approach over its state-of-the-art conventional counterpart is clearly pronounced. Similar results, not reported here, have been also obtained for the remaining designs.

As a closing remark, it is worth noting how scan chains operating in the compaction mode are handled during experiments. A fault propagating to these scan chains is temporarily marked as detected. It may not make it to the scan serial outputs because of aliasing, though. In such a case, it remains a target. Fault masking is extremely unlikely as the observation scan chains form finite memory devices [46], where after several cycles an error is shifted out. This observation is fully supported by both deferred fault crediting and the reported fault simulation experiments, which were run in the no fault dropping mode with a compaction scan chains emulator.

IX. Conclusion

In the first part of this paper, we present a DFT scheme that aims at reducing deterministic test PCs and increasing circuits' random testability. This is accomplished by deploying hybrid test points in designs where on-chip sequential test compression is combined with LBIST infrastructure. The proposed scheme reduces test application time—a crucial factor when running in-system tests for safety-critical and automotive applications. The new TPI method identifies internal conflicts that preclude efficient ATPG-based test compaction and detection of random resistant faults. Locations corresponding to such hybrid conflicts are modified by test points, which increase the number of faults targeted by a single pattern, reduce ATPG PCs and a test data volume, and finally allow one to run high-quality tests during LBIST sessions.

Subsequently, we propose an LBIST scheme that significantly reduces test time (or applies much more vectors within the same time interval) by having pseudorandom test patterns delivered in a test-per-clock fashion through conventional scan chains and by recording test results by means of per-cycle-driven hybrid observation test points that monitor the most sensitive fault propagation paths. As shown in this paper, test results are regularly saved in a per-cycle manner by means of dedicated cells and the compaction scan chains they form. The observation test point cells do not receive enable controls at speed—they have to be asserted only once to launch a test. Consequently, these signals do not have to be routed like clocks, and thus their distribution is not a primary concern.

Experimental results for large industrial designs confirm feasibility of the solutions proposed in this paper. These results were obtained for static faults, which are widely deployed for online monitoring of safety critical applications, including automotive designs. This paper does not discuss benefits of the proposed technique for delay fault coverage. In principle, however, there are no restrictions in extending the proposed methodology for transition or path-delay patterns other than adding more requirements such as closing timing for scan paths at-speed.

References

[1] C. Acero *et al.*, "Embedded deterministic test points," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2949–2961, Oct. 2017.

[2] H. Ando, "Testing VLSI with random access scan," in *Proc. COMPCON*, 1980, pp. 50–52.

[3] D. H. Baik and K. K. Saluja, "Progressive random access scan: A simultaneous solution to test power, test data volume and test time," in *Proc. ITC*, Austin, TX, USA, 2005, pp. 359–368.

[4] P. H. Bardell and W. H. McAnney, "Simultaneous self-testing system," U.S. Patent 4 513 418, Apr. 23, 1985.

[5] C. Barnhart *et al.*, "Extending OPMISR beyond 10x scan test efficiency," *IEEE Des. Test. Comput.*, vol. 19, no. 5, pp. 65–73, Sep./Oct. 2002.

[6] F. Brglez, P. Pownall, and R. Hum, "Applications of testability analysis: From ATPG to critical delay path tracing," in *Proc. ITC*, Philadelphia, PA, USA, 1984, pp. 705–712.

[7] A. J. Briers and K. A. E. Totton, "Random pattern testability by fault simulation," in *Proc. ITC*, 1986, pp. 274–281.

[8] S.-C. Chang, S.-S. Chang, W.-B. Jone, and C.-C. Tsai, "A novel combinational testability analysis by considering signal correlation," in *Proc. ITC*, Washington, DC, USA, 1998, pp. 658–667.

[9] K.-T. Cheng and C.-J. Lin, "Timing-driven test point insertion for full-scan and partial-scan BIST," in *Proc. ITC*, Washington, DC, USA, 1995, pp. 506–514.

[10] F. Corno, P. Prinetto, and M. S. Reorda, "Making the circular self-test path technique effective for real circuits," in *Proc. ITC*, Washington, DC, USA, 1994, pp. 949–957.

[11] D. Czysz *et al.*, "Deterministic clustering of incompatible test cubes for higher power-aware EDT compression," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 8, pp. 1225–1238, Aug. 2011.

[12] M. J. Geuzebroek, J. T. van der Linden, and A. J. van de Goor, "Test point insertion that facilitates ATPG in reducing test time and data volume," in *Proc. ITC*, Baltimore, MD, USA, 2002, pp. 138–147.

[13] V. Gherman *et al.*, "Efficient pattern mapping for deterministic logic BIST," in *Proc. ITC*, Charlotte, NC, USA, 2004, pp. 48–56.

[14] L. H. Goldstein and E. L. Thigpen, "SCOAP: Sandia controllability/observability analysis program," in *Proc. DAC*, Minneapolis, MN, USA, 1980, pp. 190–196.

[15] A.-W. Hakmi *et al.*, "Restrict encoding for mixed-mode BIST," in *Proc. VTS*, Santa Cruz, CA, USA, 2009, pp. 179–184.

[16] A.-W. Hakmi *et al.*, "Programmable deterministic built-in self-test," in *Proc. ITC*, 2007, paper 18.1.

[17] F. Hapke *et al.*, "Cell-aware test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 9, pp. 1396–1409, Sep. 2014.

[18] J. P. Hayes and A. D. Friedman, "Test point placement to simplify fault detection," *IEEE Trans. Comput.*, vol. C-23, no. 7, pp. 727–735, Jul. 1974.

[19] S. Hellebrand, H.-G. Liang, and H.-J. Wunderlich, "A mixed mode BIST scheme based on reseeding of folding counters," in *Proc. ITC*, Atlantic City, NJ, USA, 2000, pp. 778–784.

[20] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, "Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 223–233, Feb. 1995.

[21] S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern generation for a deterministic BIST scheme," in *Proc. ICCAD*, San Jose, CA, USA, 1995, pp. 88–94.

[22] C. Hobbs and P. Lee, "Understanding ISO 26262 ASILs," *Electron. Design*, Jul. 2013. [Online]. Available: http://www.electronicdesign.com/

[23] K. Ichino, T. Asakawa, S. Fukumoto, K. Iwasaki, and S. Kajihara, "Hybrid BIST using partially rotational scan," in *Proc. ATS*, Kyoto, Japan, 2001, pp. 379–384.

[24] V. S. Iyengar and D. Brand, "Synthesis of pseudo-random pattern testable designs," in *Proc. ITC*, Washington, DC, USA, 1989, pp. 501–508.

[25] A. Jas, C. V. Krishna, and N. A. Touba, "Weighted pseudorandom hybrid BIST," *IEEE Trans. VLSI*, vol. 12, no. 12, pp. 1277–1283, Dec. 2004.

[26] A. Jas, K. Mohanram, and N. A. Touba, "An embedded core DFT scheme to obtain highly compressed test sets," in *Proc. ATS*, Shanghai, China, 1999, pp. 275–280.

[27] E. Kalligeros, X. Kavousianos, D. Bakalis, and D. Nikolos, "An efficient seeds selection method for LFSR-based test-per-clock BIST," in *Proc. ISQED*, San Jose, CA, USA, 2002, pp. 261–266.

[28] R. Kapur, S. Mitra, and T. W. Williams, "Historical perspective on scan compression," *IEEE Des. Test. Comput.*, vol. 25, no. 2, pp. 114–120, Mar./Apr. 2008.

[29] R. Kapur, S. Patil, T. J. Snethen, and T. W. Williams, "A weighted random pattern test generation system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 8, pp. 1020–1025, Aug. 1996.
[30] B. Koenemann, "LFSR-coded test patterns for scan designs," in *Proc. ETC*, 1991, pp. 237–242.
[31] B. Koenemann, J. Mucha, and G. Zwiehoff, "Built-in logic block observation techniques," in *Proc. ITC*, 1979, pp. 37–41.
[32] H. Konuk *et al.*, "Design for low test pattern counts," in *Proc. DAC*, 2015, paper 58.4.
[33] A. Kraśniewski and S. Pilarski, "Circular self-test path: A low-cost BIST technique for VLSI circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 1, pp. 46–55, Jan. 1989.
[34] C. V. Krishna and N. A. Touba, "Hybrid BIST using an incrementally guided LFSR," in *Proc. Symp. Defect Fault Tolerance*, Boston, MA, USA, 2003, pp. 217–224.
[35] C. V. Krishna and N. A. Touba, "Reducing test data volume using LFSR reseeding with seed compression," in *Proc. ITC*, Baltimore, MD, USA, 2002, pp. 321–330.
[36] A. Kumar, J. Rajski, S. M. Reddy, and T. Rinderknecht, "On the generation of compact deterministic test sets for BIST ready designs," in *Proc. ATS*, 2013, pp. 201–206.
[37] J. Lee and N. A. Touba, "LFSR-reseeding scheme achieving low-power dissipation during test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 396–401, Feb. 2007.
[38] L. Lei and K. Chakrabarty, "Test set embedding for deterministic BIST using a reconfigurable interconnection network," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 9, pp. 1289–1305, Sep. 2004.
[39] H.-G. Liang, S. Hellebrand, and H.-J. Wunderlich, "Two-dimensional test data compression for scan-based deterministic BIST," in *Proc. ITC*, Baltimore, MD, USA, 2001, pp. 894–902.
[40] S. Milewski *et al.*, "Full-scan LBIST with capture-per-cycle hybrid test points," in *Proc. ITC*, Fort Worth, TX, USA, 2017, paper 10.3.
[41] E. Moghaddam, N. Mukherjee, J. Rajski, J. Tyszer, and J. Zawada, "Test point insertion in hybrid test compression/LBIST architectures," in *Proc. ITC*, Fort Worth, TX, USA, 2016, paper 2.1.
[42] G. Mrugalski, J. Rajski, J. Solecki, J. Tyszer, and C. Wang, "Trimodal scan-based test paradigm," *IEEE Trans. VLSI Systems*, vol. 25, no. 3, pp. 1112–1125, Mar. 2017.
[43] F. Muradali, V. K. Agarwal, and B. Nadeau-Dostie, "A new procedure for weighted random built-in self-test," in *Proc. ITC*, Washington, DC, USA, 1990, pp. 660–669.
[44] M. Nakao, K. Hatayama, and I. Highasi, "Accelerated test points selection method for scan-based BIST," in *Proc. ATS*, 1997, pp. 359–364.
[45] O. Novak and J. Nosek, "Test-per-clock testing of the circuits with scan," in *Proc. Int. Online Test. Workshop*, 2001, pp. 90–92.
[46] J. Rajski, J. Tyszer, C. Wang, and S. M. Reddy, "Finite memory test response compactors for embedded test applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 4, pp. 622–634, Apr. 2005.
[47] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Decompressor/PRPG for applying pseudo-random and deterministic test patterns," U.S. Patent 6 684 358, 2004.
[48] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. CAD*, vol. 23, no. 5, pp. 776–792, May 2004.
[49] W. Rao and A. Orailoglu, "Virtual compression through test vector stitching for scan based designs," in *Proc. DATE*, Munich, Germany, 2003, pp. 104–109.
[50] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, "Low power mixed-mode BIST based on mask pattern generation using dual LFSR re-seeding," in *Proc. ICCD*, Freiberg, Germany, 2002, pp. 474–479.
[51] B. H. Seiss, P. Trouborst, and M. Schulz, "Test point insertion for scan-based BIST," in *Proc. ETC*, 1991, pp. 253–262.
[52] R. Sethuram, S. Wang, S. T. Chakradhar, and M. L. Bushnell, "Zero cost test point insertion technique to reduce test set size and test generation time for structured ASICs," in *Proc. ATS*, Fukuoka, Japan, 2006, pp. 339–348.
[53] Y. Son, J. Chong, and G. Russell, "E-BIST: Enhanced test-per-clock BIST architecture," *IEE Proc. Comput. Digit. Techn.*, vol. 149, no. 1, pp. 9–15, Jan. 2002.
[54] C. E. Stroud, "An automated BIST approach for general sequential logic synthesis," in *Proc. DAC*, Anaheim, CA, USA, 1988, pp. 3–8.
[55] N. Tamarapalli and J. Rajski, "Constructive multi-phase test point insertion for scan-based BIST," in *Proc. ITC*, Washington, DC, USA, 1996, pp. 649–658.
[56] N. A. Touba, "Survey of test vector compression techniques," *IEEE Des. Test. Comput.*, vol. 23, no. 4, pp. 294–303, Apr. 2006.
[57] N. A. Touba and E. J. McCluskey, "Transformed pseudo-random patterns for BIST," in *Proc. VTS*, 1995, pp. 410–416.
[58] N. A. Touba and E. J. McCluskey, "Bit-fixing in pseudo-random sequences for scan BIST," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 4, pp. 545–555, Apr. 2001.
[59] H.-C. Tsai, K.-T. Cheng, C.-J. Lin, and S. Bhawmik, "A hybrid algorithm for test point selection for scan-based BIST," in *Proc. DAC*, Anaheim, CA, USA, 1997, pp. 478–483.
[60] Z. Wang and K. Chakrabarty, "Test data compression for IP embedded cores using selective encoding of scan slices," in *Proc. ITC*, Austin, TX, USA, 2005, pp. 581–590.
[61] P. Wohl, J. A. Waicukauski, S. Patel, and M. B. Amin, "Efficient compression and application of deterministic patterns in a logic BIST architecture," in *Proc. DAC*, Anaheim, CA, USA, 2003, pp. 566–569.
[62] P. Wohl *et al.*, "Efficient compression of deterministic patterns into multiple PRPG seeds," in *Proc. ITC*, Austin, TX, USA, 2005, pp. 916–925.
[63] H.-J. Wunderlich, "Multiple distributions for biased random test patterns," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 6, pp. 584–593, Jun. 1990.
[64] H.-J. Wunderlich and G. Kiefer, "Bit-flipping BIST," in *Proc. ICCAD*, San Jose, CA, USA, 1996, pp. 337–343.
[65] D. Xiang, X. Wen, and L.-T. Wang, "Low-power scan-based built-in self-test based on weighted pseudorandom test pattern generation and reseeding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 942–953, Mar. 2017.
[66] D. Xiang, Y. Xu, and H. Fujiwara, "Nonscan design for testability for synchronous sequential circuits based on conflict resolution," *IEEE Trans. Comput.*, vol. 52, no. 8, pp. 1063–1075, Aug. 2003.
[67] M. Yoshimura, T. Hosokawa, and M. Ohta, "A test point insertion method to reduce the number of test patterns," in *Proc. ATS*, 2002, pp. 298–304.
[68] F. Zhang *et al.*, "Putting wasted clock cycles to use: Enhancing fortuitous cell-aware fault detection with scan shift capture," in *Proc. ITC*, Fort Worth, TX, USA, 2016, paper 2.3.

Elham Moghaddam (S'07–M'11) received the B.S. and M.S. degrees in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2007, and the Ph.D. degree in computer and electronic engineering from the University of Iowa, Iowa City, IA, USA, in 2011.

She is currently a Software Developer with the Design-to-Silicon Division, Mentor, A Siemens Business, Wilsonville, OR, USA. Her current research interests include design for testability, low power embedded test, built-in self-test, and test data compression.

Nilanjan Mukherjee (S'87–M'89–SM'14) received the B.Tech. degree (Hons.) in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1989 and the Ph.D. degree from McGill University, Montreal, QC, Canada, in 1996.

He is currently an Engineering Director with the Design-to-Silicon division at Mentor, A Siemens Business, Wilsonville, OR, USA. He is a Co-Inventor of the EDT technology and was a Lead Developer for the leading test compression tool in the industry, TestKompress, Wilsonville, OR, USA. His current research interests include next generation test methodologies for deep sub-micron designs, test data compression, test synthesis, memory testing, and fault diagnosis. He has published over 75 technical papers and has co-invented 45 U.S. patents.

Dr. Mukherjee was a co-recipient of the Best Paper Award at the 1995 IEEE VLSI Test Symposium, the Best Paper Award at the 2009 VLSI Design Conference, the 2006 IEEE Circuits and Systems Society Donald O. Pederson Outstanding Paper Award recognizing this paper on embedded deterministic test published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the 2012 IEEE International Test Conference Most Significant Paper Award. He served on the program committees of several IEEE conferences.

Janusz Rajski (A'87–SM'10–F'11) received the M.S. degree in electrical engineering from the Technical University of Gdańsk, Gdańsk, Poland, in 1973 and the Ph.D. degree in electrical engineering from the Poznań University of Technology, Poznań, Poland, in 1982.

From 1973 to 1984, he was a Faculty Member of the Poznań University of Technology. From 1984 to 1995, he was with McGill University, Montreal, QC, Canada, In 1995, he accepted the position of Chief Scientist with Mentor Graphics, Wilsonville, OR, USA. He is currently the Vice President of Engineering at Mentor, A Siemens Business, Wilsonville, OR, USA. He is also the Principal Inventor of the Embedded Deterministic Test technology used in the first commercial test compression product TestKompress, Wilsonville, OR, USA. His current research interests include testing of very large scale integration systems, design for testability, built-in self-test, and logic synthesis. He has published over 200 research papers in these areas and has co-invented 107 U.S. patents.

Dr. Rajski was a co-recipient of the 1993 Best Paper Award for this paper on logic synthesis published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the 1995 and 1998 Best Paper Awards at the IEEE VLSI Test Symposium, the 1999 and 2003 Honorable Mention Awards at the IEEE International Test Conference, as well the 2006 IEEE Circuits and Systems Society Donald O. Pederson Outstanding Paper Award recognizing this paper on embedded deterministic test published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the 2009 Best Paper Award at the VLSI Design Conference, the 2011 Best Paper Award at the IEEE European Test Symposium, and the 2012 IEEE International Test Conference Most Significant Paper Award. He has served on technical program committees of various conferences.

Jędrzej Solecki (M'15) received the M.S. degree in computer science and the Ph.D. degree in electrical engineering from the Poznań University of Technology, Poznań, Poland, in 2011 and 2015, respectively.

Since 2015, he has been with Mentor, A Siemens Business, Wilsonville, OR, USA. His current research interests include design for testability, built-in self-test, embedded test, and automatic test pattern generation.

Dr. Solecki was a co-recipient of the 2011 Best Paper Award at the IEEE European Test Symposium.

Jerzy Tyszer (M'91–SM'96–F'13) received the M.S. and Ph.D. degrees in electrical engineering from the Poznań University of Technology, Poznań, Poland, in 1981 and 1987, respectively, and the Dr.Hab. degree in telecommunications from the Technical University of Gdańsk, Gdańsk, Poland, in 1994.

From 1982 to 1990, he was a Faculty Member of the Poznań University of Technology. In 1990, he joined McGill University, Montreal, QC, Canada, where he was a Research Associate and an Adjunct Professor. In 1996, he assumed the position of a Professor with the Faculty of Electronics and Telecommunications, Poznań University of Technology, Poznań, Poland. His current research interests include design automation and testing of very large scale integration systems, design for testability, built-in self-test, embedded test, and computer simulation of discrete event systems. He has published eight books, over 140 research papers in the above areas and has co-invented 71 U.S. patents.

He was a co-recipient of the 1995 and 1998 Best Paper Awards at the IEEE VLSI Test Symposium, the 2003 Honorable Mention Award at the IEEE International Test Conference, the 2006 IEEE Circuits and Systems Society Donald O. Pederson Outstanding Paper Award recognizing this paper on embedded deterministic test published in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the 2009 Best Paper Award at the VLSI Design Conference, the 2011 Best Paper Award at the IEEE European Test Symposium, and the 2012 IEEE International Test Conference Most Significant Paper Award. He has served on technical program committees of various conferences.

Justyna Zawada (M'14) received the M.S. degree in computer science from Adam Mickiewicz University, Poznań, Poland, in 2012, the M.S. degree in telecommunications and the Ph.D. degree in electrical engineering from the Poznań University of Technology, Poznań, in 2014 and 2017, respectively.

She is currently a Software Developer with the Design-to-Silicon Division, Mentor, A Siemens Business, Wilsonville, OR, USA. Her current research interests include design for testability, built-in self-test, embedded test, automatic test pattern generation, and test security.