Layout Resynthesis by Applying Design-for-manufacturability Guidelines to Avoid Low-coverage Areas of a Cell-based Design

NAIXING WANG and IRITH POMERANZ, Purdue University, USA
SUDHAKAR M. REDDY, University of Iowa, USA
ARANI SINHA and SRIKANTH VENKATARAMAN, Intel Corporation, USA

Design-for-manufacturability (DFM) guidelines are recommended layout design practices intended to capture layout features that are difficult to manufacture correctly. Avoiding such features prevents the occurrence of potential systematic defects. Layout features that result in DFM guideline violations may not be avoided completely due to the design constraints of chip area, performance, and power consumption. A framework for translating DFM guideline violations into potential systematic defects, and faults, was described earlier. In a cell-based design, the translated faults may be internal or external to cells. In this article, we focus on undetectable faults that are external to cells. Using a resynthesis procedure that makes fine changes to the layout while maintaining the design constraints, we target areas of the design where large numbers of external faults related to DFM guideline violations are undetectable. By eliminating the corresponding DFM guideline violations, we ensure that the circuit does not suffer from low-coverage areas that may result in detectable systematic defects escaping detection, but failing the circuit in the field. The layout resynthesis procedure is applied to benchmark circuits and logic blocks of the OpenSPARC T1 microprocessor. Experimental results indicate that the improvement in the coverage of potential systematic defects is significant.

CCS Concepts: • **Hardware → Defect-based test**; **Test-pattern generation and fault simulation**; *VLSI design manufacturing considerations*; Physical synthesis;

Additional Key Words and Phrases: Design-for-manufacturability (DFM) guidelines, layout resynthesis, systematic defects, undetectable faults

ACM Reference format:
Naixing Wang, Irith Pomeranz, Sudhakar M. Reddy, Arani Sinha, and Srikanth Venkataraman. 2019. Layout Resynthesis by Applying Design-for-manufacturability Guidelines to Avoid Low-coverage Areas of a Cell-based Design. *ACM Trans. Des. Autom. Electron. Syst.* 24, 4, Article 42 (May 2019), 19 pages.
https://doi.org/10.1145/3325066

The work of N. Wang and I. Pomeranz was supported in part by NSF Grant No. CCF-1714147.
Authors' addresses: N. Wang and I. Pomeranz, Purdue University, School of Electrical and Computer Engineering, West Lafayette, IN 47907; emails: {wang2489, pomeranz}@purdue.edu; S. M. Reddy, University of Iowa, Electrical and Computer Engineering Department, Iowa City, IA 52242; email: sudhakar-reddy@uiowa.edu; A. Sinha and S. Venkataraman, Intel Corporation, Hillsboro, OR 97124; emails: {arani.sinha, srikanth.venkataraman}@intel.com.

1084-4309/2019/05-ART42 $15.00
https://doi.org/10.1145/3325066

1 INTRODUCTION

The scaling of integrated circuit (IC) technologies has brought about many benefits, including faster devices, lower power consumption, reduced chip sizes, and increase in functionality. However, the continuous shrinking of device sizes also increases the gap between the feature size and the lithography wavelength. As a result, for each smaller process technology node, the chips are increasingly impacted by deviations in manufactured patterns from the intended design. Specifically, certain layout features are more difficult to manufacture than others and are more likely to lead to circuit failures. When such features are present multiple times in a chip, they can result in repeated or systematic defects, which can impact the yield and defective-parts-per-million (DPPM) significantly [5, 17, 18, 32, 33, 38]. Due to modeling errors and algorithmic inaccuracies in removing the resulting systematic variations, process-related corrective actions using OPC/RET techniques are not sufficient for acceptable yield and DPPM [3]. Thus, appropriate interventions during circuit design are inevitable to remedy the potential manufacturing issues and address the systematic defects.

Such design interventions are formulated as design rules and design-for-manufacturability (DFM) guidelines. While design rules are mandatory, and must all be applied to a design, DFM guidelines are taken as recommendations, and they are adhered to when possible within the design constraints of area, performance, and power consumption. When DFM guidelines are not adhered to, potential systematic defects may occur. The relationship between DFM guideline violations and potential systematic defects was discussed in References [13, 14, 34]. In Reference [13], DFM guidelines related to vias on interconnects, and contacts on p-diffusion, are considered. A more comprehensive set of DFM guidelines is considered in Reference [14]. DFM guidelines related to internal nets of standard cells are considered in Reference [34]. In all these works, the layout sites where DFM guidelines are violated are found, and the affected transistors are identified at the schematic level. The anticipated defect behaviors are then translated into gate-level logic faults using switch-level simulation. Test generation is carried out for the resulting faults to avoid potential test holes.

Among the potential faults resulting from DFM guideline violations, there are undetectable faults. When a large number of undetectable faults related to DFM guideline violations are present in an area of the circuit, the area suffers from a low coverage by tests that target the area. The missing tests may allow detectable defects in the area to escape detection. This can impact the yield and reliability significantly, since the defects are likely to be systematic. This article demonstrates these issues and addresses them in the context of a cell-based design, and targeting faults that are external to cells. For this discussion, we distinguish between faults that are internal and ones that are external to cells. We assume that undetectable faults, which are related to DFM guideline violations, and are internal to cells, can be addressed by proper logic synthesis [8, 22, 39]. In Reference [8, 22], manufacturability information is integrated into the cost function of logic synthesis, and a DFM extension library that contains yield-optimized cells is used for improving the manufacturability of the circuit. In Reference [39], a logic resynthesis procedure is proposed that replaces cells with large numbers of undetectable internal faults related to DFM guideline violations by smaller cells that contain fewer faults. Overall, the procedure eliminates undetectable faults that are internal to cells, and also reduces the number of undetectable external faults. However, it leaves large numbers of undetectable external faults that cannot be addressed by replacing cells.

For faults that are external to cells, this article describes a layout resynthesis procedure that makes fine changes to the layout while maintaining the design constraints to improve the coverage of areas with low coverage because of the presence of undetectable external faults. The layout resynthesis procedure in itself (the procedure that makes local changes to the layout) is not the main contribution of the article. The contribution is related to the use of DFM guidelines to identify

areas of the circuit with low coverage, and improving their coverage by layout resynthesis. From a test point of view, we show which DFM guideline violations need to be considered first to improve the layout areas with low coverage. This is the first layout resynthesis procedure to address this issue directly. As part of this solution, we suggest a layout-based coverage metric that can be used for identifying areas with low coverage.

As technologies evolve, many DFM guidelines remain the same and are transferred to the new technology. For example, the interconnect bridge defects shown in Reference [20] to exist in 160nm technology are also believed to be an issue in the latest FinFET technologies [10], and require similar DFM guidelines. The proposed layout resynthesis procedure is independent of the cell library and the DFM guidelines. Therefore, it can work with different DFM guidelines related to different technology nodes even if new DFM guidelines are introduced. For different DFM guidelines, it may require different changes to the layout for fixing DFM guideline violations, but the basic methodology is the same.

The proposed layout resynthesis procedure eliminates undetectable external faults by fixing the DFM guideline violations that lead to them. The procedure prefers to eliminate faults whose effect on the coverage of the circuit is more significant. The DFM guideline violations are fixed by automatically changing the layout with the help of a place and route tool. The procedure does not allow any increase in critical path delay, power consumption or die area when changing the layout.

Other approaches for addressing the testability of a circuit are described in References [1, 4, 6, 9, 15, 16, 19, 21, 24–28, 31, 40–44]. In References [6, 25, 26, 28, 40], design-for-testability (DFT) methods for improving the transition fault coverage are discussed. The coverage is improved by inserting DFT logic that can provide better control over the state vectors. In References [4, 16, 24, 27, 31, 43], logic resynthesis techniques are used for improving the testability of the circuit by reducing the difficulty of test generation. In References [9, 41, 42], scan chain ordering is considered for improving the coverage for transition and path delay faults. With layout information taken into account, the routing penalty and the impact on circuit performance are limited. In References [1, 15, 19, 21, 44], test point insertion is considered for improving the testability of a circuit, while limiting the deterministic pattern counts. We experimented with circuits into which test points are inserted to improve testability. The results indicate that, even with test points, there are areas with low coverage for faults that result from DFM guideline violations. Such areas require the layout resynthesis procedure described in this article. The reason is that test point insertion does not target DFM guideline violations directly, and therefore, does not target the resulting undetectable faults.

The procedure described in this article can be embedded into a standard cell-based design flow. In a cell-based design flow, after the initial design of the layout, several iterations of an incremental physical design process are typically required for satisfying the design constraints of delay, power and area. The proposed procedure is also iterative, and can thus fit within the overall iterative design process. In particular, an iteration of the design process can include one or more iterations of the proposed procedure to eliminate undetectable faults in poorly covered circuit areas, and improve the coverage of potential systematic defects. For a large chip, to maintain an acceptable computational effort, the proposed procedure can be applied to each logic block separately. We applied the proposed procedure to logic blocks of the OpenSPARC T1 microprocessor to demonstrate its applicability to such designs.

This article is organized as follows. Section 2 demonstrates the existence of undetectable external faults related to DFM guideline violations, and the presence of areas with poor coverage. Section 3 describes the proposed layout resynthesis procedure. Experimental results and analysis are presented in Section 4.

Table 1. DFM Guideline Violations

Circuit	DFM_total	DFM_undet
b15	218,619	10,786
b20	249,383	3,352
sparc_fpu	1,666,799	9,012
sparc_exu	899,740	20,328

2 UNDETECTABLE FAULTS RELATED TO DFM GUIDELINE VIOLATIONS

This section discusses the existence of undetectable external faults related to DFM guideline violations, and the presence of areas with poor coverage. A coverage metric is defined based on layout neighborhoods of undetectable faults.

Three categories of DFM guidelines are considered in this article. They are *Via*, *Metal* and *Density*. We use 19 guidelines in the *Via* category, 29 guidelines in the *Metal* category, and 11 guidelines in the *Density* category. These guidelines provide recommended layout constraints for dimensions of vias, spacings between exterior-facing edges on polygons, and densities of routing layers.

We translate DFM guideline violations into potential short and open defects that are external to cells, and then translate the potential defects into corresponding logic faults using the approach described in References [13, 14, 34]. We denote the set of faults by F. A test generation procedure is applied to generate a test set T that detects all the detectable faults in F. We denote by $U = \{f_1, f_2, \ldots, f_n\}$ the set of undetectable faults in F.

2.1 Analysis of DFM Guideline Violations

In this section, we discuss the challenges related to DFM guidelines.

In Table 1, we show the numbers of DFM guideline violations for several circuits. In column *DFM_total*, we show the total number of DFM guideline violations in the circuit. In column *DFM_undet*, we show the number of DFM guideline violations translated to undetectable faults. It can be observed that the number of DFM guideline violations is typically very large, and it is not possible to fix all of them within the design constraints. The number of DFM guideline violations translated to undetectable faults is small compared to the total number of DFM guideline violations. This makes it possible for the layout resynthesis procedure described in this article to address them.

2.2 Detectable Defects Modeled by Undetectable Faults

In this section, we consider an example where faults that are translated from DFM guideline violations are undetectable, while potential systematic defects in the same area are detectable. The presence of such situations motivates the need for layout resynthesis to eliminate undetectable faults and improve the coverage of the area.

We conducted the following experiment to determine the existence of detectable, potentially systematic defects that may go undetected. The DFM guideline we considered specifies the recommended minimum separation between exterior facing edges of *metal4* polygons. Figure 1 shows an example where this DFM guideline is violated for NET1 and NET2, as well as NET2 and NET3. These violations can potentially cause shorts between NET1 and NET2, and between NET2 and NET3. The shorts are modeled by bridging faults. Suppose that both bridging faults are undetectable.

A possible defect that is not covered by the DFM guideline is a short between NET1 and NET3 (i.e., NET1, NET2, and NET3 are shorted). This defect is not modeled by F, and a test for it is not

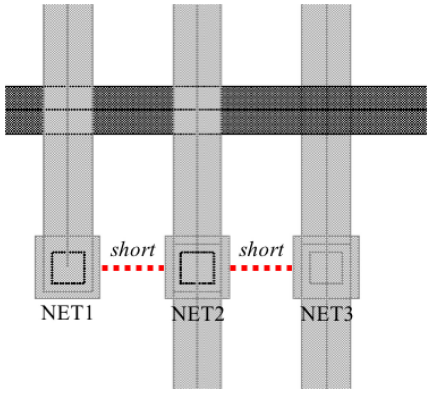

Fig. 1. Potential short defects.

Table 2. Uncovered Short Defects

Circuit	Uncov.
b15	42
b20	65
sparc_fpu	75
sparc_exu	76

generated directly. Nevertheless, the defect may occur because the site is prone to be defective. Without tests that detect the shorts between NET1 and NET2, and between NET2 and NET3, this defect may remain undetected by *T*. In this case, the low coverage around NET1, NET2, and NET3, and the missing tests for the two bridging faults, causes the bridge between NET1 and NET3 to go undetected.

We searched for occurrences of this situation in benchmark circuits and logic blocks of the OpenSPARC T1 microprocessor. The test set we used detects all the detectable transition and stuck-at faults, as well as the bridging faults in F. The results for several circuits are shown in Table 2. For every circuit, column *Uncov* shows the number of occurrences of undetected defects as illustrated by Figure 1.

Although the numbers in Table 2 are small, they represent only one example where the presence of undetectable faults may allow detectable defects to go undetected. This motivates the layout resynthesis procedure described in Section 3 that eliminates undetectable faults in areas with low coverage.

In general, since sites of DFM guideline violations are more likely to be defective than other sites, and circuits manufactured prior to volume production tend to suffer from multiple defects, it can be expected that multiple DFM violation sites would be defective. In addition to the double fault illustrated above, there can be other types of undetectable faults related to DFM guideline violations that are undetectable alone, but become detectable when two or more faults are present together. One can try to add tests to detect multiple faults [7, 12, 29, 30], but the number of faults can be very large, and the number of tests may increase dramatically. For example, in the circuits we considered, we found hundreds and even thousands of undetectable faults related to DFM guideline violations, leading to millions and more multiple faults. The resynthesis procedure we propose in this article eliminates or drastically reduces the number of undetectable faults related

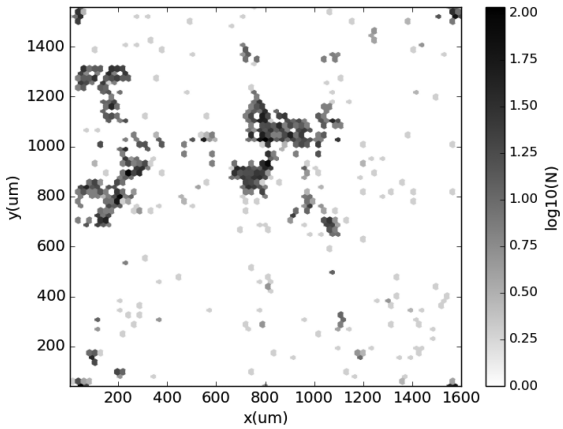

Fig. 2. Undetectable external faults related to DFM guideline violations in *sparc_fpu*.

to DFM guideline violations. If desired, one can add tests for detectable multiple faults that consist of undetectable faults remaining after resynthesis [29].

2.3 Circuit Areas with Poor Coverage

In this section, we define a coverage metric, and demonstrate the presence of areas with poor coverage that suffer from the presence of undetectable external faults related to DFM guideline violations.

In Figure 2, we show the topological distribution of the undetectable external faults related to DFM guideline violations in the layout of *sparc_fpu*. It can be observed that the undetectable faults tend to cluster in the darker areas, resulting in areas with large numbers of undetectable faults.

We define the coverage of an area as the percentage of detectable faults among all the faults related to DFM guideline violations in this area. The details of this definition are discussed next.

We say that two layout sites are adjacent to each other if the distance between them is less than 20× the minimum feature size. Within such distance, the two layout sites can be affected by similar optical interactions, which are the major causes for systematic defects [11].

For a defect d that is obtained from a DFM guideline violation, we define the neighborhood of d to include all the layout sites that are adjacent to the site of d.

A fault f in F may model several different defects. The defects are always at a close proximity to each other. We define the neighborhood of f as the union of the neighborhoods of all the defects that f models.

We say that a fault f' is in the neighborhood of a fault f if the site of a defect d' modeled by f' is in the neighborhood of f. With these definitions, we define the coverage $c(f)$ of the neighborhood of a fault f as follows:

$$c(f) = \frac{Number\ of\ detectable\ faults\ in\ the\ neighborhood\ of\ f}{Total\ number\ of\ faults\ in\ the\ neighborhood\ of\ f}. \quad (1)$$

For illustration, we computed coverages for the neighborhoods of all the undetectable external faults in *sparc_fpu*, and we partitioned the faults according to their coverage range. The results are shown in Figure 3.

It can be seen that the coverages for more than half of the undetectable external faults are below 90%. About a quarter of the faults have coverages below 60%. These observations motivate the need

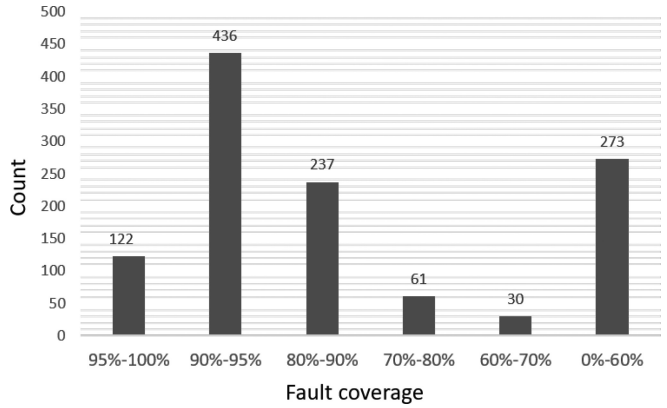

Fig. 3. Coverages for the undetectable external faults in *sparc_fpu*.

to eliminate undetectable external faults with poorly covered neighborhoods and thus improve the coverage for potential systematic defects.

2.4 Coverage for Faults with Weighted DFM Guidelines

The coverage metric $c(f)$ defined above assumes that all the DFM guidelines are equally important. In this section, we define the coverage of the neighborhood of a fault f when DFM guidelines have different levels of importance.

The evaluation of DFM guidelines was discussed in References [2, 36, 37]. In Reference [36], test structures are used for evaluating the importance of DFM guidelines. In References [2, 37], information extracted from actual failed ICs during volume diagnosis is used for measuring the effectiveness of a DFM guideline.

In this article, we define the weight of a DFM guideline as the probability of a defect given a violation of this guideline. A violation of a DFM guideline with a higher weight indicates a higher risk of failure. We also define the weight of a fault related to DFM guideline violations based on the corresponding DFM guideline weights, as follows.

Suppose that a fault f is translated from violations of n different DFM guidelines, $g_1, g_2, \ldots, g_n$, and the weights of $g_1, g_2, \ldots, g_n$ are $p_1, p_2, \ldots, p_n$. The numbers of times that $g_1, g_2, \ldots, g_n$ are violated are denoted by $m_1, m_2, \ldots, m_n$. We assume that all the DFM guideline violations are independent. We define the weight $w(f)$ of f as the probability that f is present in the circuit given all the DFM guideline violations that are translated to f. We have that

$$w(f) = 1 - (1 - p_1)^{m_1}(1 - p_2)^{m_2}\ldots(1 - p_n)^{m_n}. \tag{2}$$

Next, considering all the faults in the neighborhood of f, we define the coverage $c(f)$ of the neighborhood of f as shown in Equation (3). In Equation (3), $\sum w(det)$ and $\sum w(undet)$ give the total weights of the detectable and undetectable faults in the neighborhood of f, respectively. In addition, they can be considered as the expected numbers of detectable and undetectable faults in the neighborhood of f. Thus, the subtrahend of the equation can be considered as the probability that f is present in the circuit, while its neighborhood is not covered by the test set that detects all the detectable faults. When this probability is high, the detectable systematic defects in the neighborhood of f may go undetected, causing circuit failure:

$$c(f) = 1 - w(f) \cdot \left(1 - \frac{\sum w(det)}{\sum w(det) + \sum w(undet)}\right). \tag{3}$$

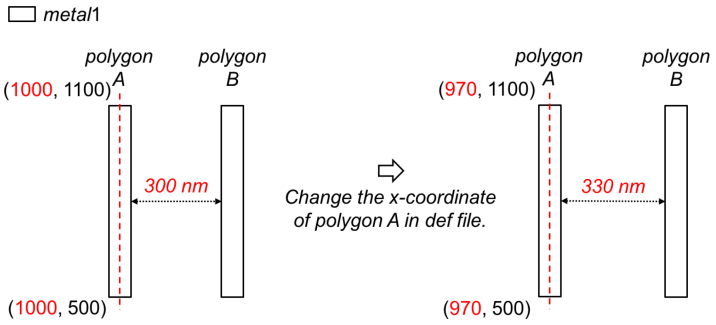

Fig. 4. Fixing a DFM guideline violation.

3 LAYOUT RESYNTHESIS

This section describes a methodology for eliminating undetectable faults by fixing the DFM guideline violations leading to them, and the layout resynthesis procedure that is built upon it.

3.1 Fixing DFM Guideline Violations

In this section, we use an example to illustrate a methodology for eliminating the undetectable faults related to DFM guideline violations. This methodology is based on the use of a place and route tool to make local changes to the layout for fixing the related DFM guideline violations. For different DFM guideline violations, the concrete modifications to the layout may be different. The connectivity of the circuit is maintained when modifying the layout as described later. The DFM guideline used as an example specifies that the separation between exterior facing edges of *metal1* polygons should be no less than 330nm. In Figure 4, this DFM guideline is violated for polygons *A* and *B*, resulting in an undetectable bridging fault that involves the two polygons.

One way to fix this violation is to move polygon *A* horizontally, such that the separation between the polygons would be enlarged. We achieve this by first extracting the design exchange format (def) file of the layout. The def file of a layout records the coordinates of all the polygons in the layout. We then change the x-coordinate of polygon *A* by subtracting 30 units. We also change the x-coordinates of the polygons that are connected to polygon *A* in the same manner to ensure the connectivity of the circuit. After modifying the def file, we provide the modified def file back to the place and route tool. The tool changes the locations of polygon *A* and polygons that are connected to polygon *A* in the layout accordingly. After moving polygon *A* to the new layout site, the separation between exterior facing edges of polygons *A* and *B* is 330nm, which adheres to the DFM guideline considered. As a result, the DFM guideline violation that results in the undetectable bridging fault is eliminated.

When an undetectable fault is translated from several different DFM guideline violations, we attempt to eliminate the undetectable fault completely to improve the coverage of its neighborhood. This is achieved by fixing all the DFM guideline violations leading to the fault, and changing all the polygons involved.

The resynthesis procedure sometimes decides to undo a layout modification. To implement this operation, the procedure stores the original coordinates of the polygons that it moves. By changing the coordinates of a polygon in the def file to its original coordinates, the polygon is moved back to its original position.

3.2 Layout Resynthesis Procedure

In this section, we describe an iterative layout resynthesis procedure that makes fine changes to the layout to eliminate undetectable faults whose neighborhoods have low coverage. This is

achieved by fixing the DFM guideline violations that lead to them. The procedure considers every undetectable fault, and attempts to eliminate the ones whose neighborhoods have the lowest coverages. When an undetectable fault is eliminated from a neighborhood with a low coverage, the coverage of the neighborhood increases.

The proposed procedure starts with the original physical design L_{ayout} of the circuit. We assume that L_{ayout} was already optimized by one or more iterations of a physical design flow, and satisfies the design constraints of delay, power and area. The proposed procedure improves L_{ayout} with respect to the coverage of the circuit without violating the original design constraints.

We use a set U_{done} to store every undetectable fault, for which the procedure has completed the attempt to eliminate it successfully. Initially, U_{done} is empty. The target coverage of the original neighborhood of an undetectable fault is denoted by p. The proposed procedure attempts to ensure that the coverage of the original neighborhood of every undetectable fault initially in the circuit is no less than p. A higher value of p indicates a higher coverage for potential systematic defects after applying the procedure, and more changes to the layout that require a higher runtime. The proposed layout resynthesis procedure can accommodate different values of p. With $p = 100\%$, the procedure considers all the undetectable faults.

In every iteration, the procedure computes the set U_{cur} of undetectable faults resulting from DFM guideline violations that are not in U_{done} based on the current layout of the circuit. For every fault it also computes its coverage. A fault with a coverage of p and above is excluded from U_{cur}. The procedure attempts to eliminate the undetectable faults in U_{cur} one by one, considering the faults from low to high coverage of their neighborhoods. This ensures that the faults with the lowest coverages are considered earlier. Such faults are also more important to consider.

When an undetectable fault f from U_{cur} is considered, it is possible that another fault, f', in its neighborhood has already been considered and added to U_{done} in this iteration. In this case, the procedure does not consider f. The reason is that the elimination of f' made a change to its neighborhood that may affect the coverage for f. In the next iteration, the procedure will recompute the coverage for f, and decide whether it still needs to be considered.

After every attempt to eliminate an undetectable fault, the procedure checks whether the design constraints of delay, power and area are satisfied. It also checks whether the modified design has Design-Rule-Check (DRC) or Layout-Versus-Schematic (LVS) violations. If all the design constraints are satisfied and there is no DRC or LVS violation, then the undetectable fault is added to U_{done}. Otherwise, the modification to the layout is discarded, and the next eligible undetectable fault in U_{cur} is considered by the procedure.

After considering all the eligible undetectable faults in U_{cur}, and modifying the layout accordingly, the proposed procedure recomputes the faults related to DFM guideline violations based on the modified layout. A test generation procedure for fault detection is carried out only when new faults are obtained. The procedure then computes the coverages of the original neighborhoods of the undetectable faults targeted in this iteration. For every target undetectable fault f, the procedure checks whether the coverage of the original neighborhood of f increased, and the layout modification for eliminating f does not result in more DFM guideline violations leading to undetectable faults. If these conditions are not satisfied, then the layout modification for fixing the DFM guideline violations leading to f is discarded.

The procedure described above is shown in Algorithm 1. We denote the process for fixing the DFM guideline violations leading to a fault f by $ApplyDFM(f)$, and the reverse process for removing the corresponding layout modification by $UnapplyDFM(f)$. The procedure terminates when (1) the coverages of all the original neighborhoods of the undetectable faults initially in the circuit are at least p, or (2) these coverages cannot be improved further without violating the design

ALGORITHM 1: Layout Resynthesis Procedure

$U_{done} = \varnothing$;
L_{ayout} is the original layout of the circuit;
repeat
  Compute U_{cur} based on L_{ayout}, p and U_{done};
  if $U_{cur} == \varnothing$ **then**
    stop
  end
  Sort the faults in U_{cur} in the order of increasing neighborhood coverages;
  for every fault f in U_{cur}, if it is eligible **do**
    $layout = ApplyDFM(f)$;
    if the design constraints are satisfied and there is no DRC or LVS violation **then**
      $L_{ayout} = layout$;
      Add f into U_{done};
    end
  end
  if L_{ayout} is not modified **then**
    stop
  end
  for every fault f added into U_{done} in this iteration, if the coverage of its original neighborhood does not increase or more DFM guideline violations causing undetectable faults are obtained **do**
    $L_{ayout} = UnapplyDFM(f)$;
  end

constraints of delay, power and area, or introducing more DFM guideline violations leading to undetectable faults.

4 EXPERIMENTAL RESULTS

In this section, we describe five experiments to demonstrate the effectiveness and applicability of the proposed layout resynthesis procedure in different scenarios.

In all the experiments, we apply the proposed procedure to benchmark circuits, and to logic blocks of the OpenSPARC T1 [23] microprocessor. OpenSPARC T1 is a 64-bit open-source microprocessor. It has eight cores. Each core can support up to four threads for a total of 32 threads. The proposed procedure is applied to the logic blocks in a single SPARC core, and the floating-point unit (*sparc_fpu*).We run the procedure on a Linux machine with 2.6GHz processors.

We use the tool kit with a standard cell library developed by OSU [35] to synthesize the RTL descriptions of the circuits into gate-level netlists and layouts. The tool kit was developed based on TSMC 180nm technology. For the circuits considered in all the experiments, the netlists obtained after logic synthesis are flattened and treated as one block with respect to floorplanning. In the first four experiments, we set the cell utilization to be 70% for all the circuits. The allocation area, total number of nets and total wire length of the original layout for each circuit are shown in Table 3. In the fifth experiment, we set the cell utilization to be 80% and 90% to show the applicability of the proposed procedure to more congested designs.

We use a commercial tool for logic synthesis. We also use a commercial place and route tool for layout synthesis and for applying DFM guidelines. A commercial IC verification and sign-off package is used for finding locations of DFM guideline violations in the layout. We use a commercial automatic test pattern generation (ATPG) tool to generate test patterns for fault detection.

Table 3. Layouts with 70% Cell Utilization

Circuit	alloc_area (mm^2)	nets	wire_length (um)
b14	0.15	17,480	139,766
b15	0.28	33,640	276,498
b20	0.32	38,049	291,774
aes_core	0.7	119,197	1,143,766
DMA	0.75	78,925	915,701
tv80	0.25	32,988	290,335
systemcaes	0.37	46,692	517,923
s35932	0.67	47,543	401,998
wb_conmax	1.1	225,595	3,092,420
sparc_spu	0.6	49,786	554,713
sparc_ffu	0.51	56,240	617,818
sparc_exu	1.09	140,785	1,992,839
sparc_lsu	1.99	213,751	3,403,376
sparc_tlu	1.94	196,218	2,743,798
sparc_ifu	1.59	187,349	2,653,452
sparc-fpu	2.33	254,847	2,759,603

4.1 Layout Resynthesis Procedure

In this experiment, the layout resynthesis procedure as described in Section 3 is applied. The results are shown in Tables 4 and 5 as follows. In each case, three rows correspond to a circuit. The first row describes the original design of the circuit. Row *half* describes the resynthesized circuit when p, the target coverage of the original neighborhood of an undetectable fault, is set to be the median of the coverages of the neighborhoods of all the undetectable faults in the original design. This indicates that only half of the original undetectable faults are considered for elimination. Row *all* describes the resynthesized circuit when p is set to be 100%, which indicates that the procedure attempts to eliminate all the undetectable faults related to DFM guideline violations.

In every row, column F provides the total number of faults related to DFM guideline violations. Column U provides the number of undetectable faults in F. Column *Cov* provides the coverage of the circuit, which is defined as $Cov = (1 - U/F)\%$. Column *Ave_c* provides the average coverage of the original neighborhoods of undetectable faults initially in the circuit. Column *Nchanges* provides the number of changes to polygons during layout resynthesis. In column *Rtime*, we show the run time for the proposed layout resynthesis procedure relative to the run time for one iteration of logic synthesis and physical design with test generation for the logic faults related to DFM guideline violations. We include test generation time, since a test set is also obtained by the proposed layout resynthesis procedure.

From Tables 4 and 5 it can be observed that the procedure described in this article achieves a significant reduction in the numbers of undetectable faults for all the circuits considered. This can be seen from column U. As the proposed procedure always attempts to eliminate the undetectable faults with the lowest coverages, and no additional DFM guideline violations causing undetectable faults are allowed to be introduced to the circuit, the coverage of the original neighborhoods of undetectable faults increases significantly. This can be seen from column *Ave_c*.

It can also be observed that when p is set to be the median of the coverages of the neighborhoods of undetectable faults in the original design, it typically requires less than half of the number of polygon changes and runtime relative to the case where p is set to be 100%. This is because

Table 4. Layout Resynthesis (Benchmarks)

Circuit		F	U	Cov	Ave_c	Nchanges	Rtime
b14	orig	11,582	182	98.43%	89.14%	/	1
	half	11,518	117	98.98%	94.02%	189	2.02
	all	11,404	2	99.98%	99.87%	556	5.7
b15	orig	21,103	713	96.62%	85.64%	/	1
	half	20,807	417	98.00%	93.07%	631	3.28
	all	20,408	10	99.95%	99.80%	2398	7.02
b20	orig	25,815	339	98.69%	91.00%	/	1
	half	25,704	213	99.17%	95.14%	329	2.12
	all	25,492	2	99.99%	99.93%	1037	6.36
aes_core	orig	75,092	874	98.84%	93.93%	/	1
	half	74,688	455	99.39%	97.49%	575	4.18
	all	74,240	11	99.99%	99.90%	1211	9.5
DMA	orig	44,589	409	99.08%	91.55%	/	1
	half	44,425	231	99.48%	96.23%	286	2.66
	all	44,191	2	99.99%	99.95%	821	6.63
tv80	orig	20,292	757	96.27%	80.43%	/	1
	half	20,048	508	97.47%	87.89%	506	1.74
	all	19,542	3	99.99%	99.95%	2618	6.77
systemcaes	orig	26,214	267	98.98%	89.43%	/	1
	half	26,102	146	99.44%	95.64%	238	1.95
	all	25,958	2	99.99%	99.87%	1240	3.52
s35932	orig	32,895	227	99.31%	89.46%	/	1
	half	32,804	122	99.63%	95.83%	69	2.59
	all	32,676	0	100.00%	100.00%	194	5.37
wb_conmax	orig	112,351	627	99.44%	93.98%	/	1
	half	112,047	319	99.72%	97.82%	340	2.22
	all	111,756	19	99.98%	99.82%	785	3.38

eliminating an undetectable fault typically increases the coverages for the undetectable faults in its neighborhood. Thus, fewer than half of the undetectable faults need to be considered.

From column *Rtime*, we can see that the relative runtime does not increase as the complexity of the circuit increases. This is because the layout modification for applying DFM guidelines is based on the original layout of the circuit. The proposed procedure does not require the layout to be implemented from the gate-level netlist repeatedly.

For further illustration, in Figure 5, we show the numbers of the original neighborhoods of undetectable faults initially in the circuit in different coverage ranges for *sparc_fpu*. The three groups of bars correspond to the three rows for *sparc_fpu* in Table 5. It can be observed that the coverages of the original neighborhoods of undetectable faults increase significantly.

4.2 Circuits with Test Points

In this section, we show the experimental results of applying the layout resynthesis procedure to circuits with test points.

The commercial tool we use inserts test points for three purposes, (1) to improve coverage for random pattern resistant faults, (2) to improve coverage for undetected faults during ATPG, and

Table 5. Layout Resynthesis (OpenSPARC T1)

Circuit		F	U	Cov	Ave_c	Nchanges	Rtime
sparc_spu	orig	30,046	295	99.02%	71.34%	/	1
	half	29,926	174	99.42%	88.38%	109	2
	all	29,790	24	99.92%	94.64%	562	4.95
sparc_ffu	orig	32,605	413	98.73%	74.05%	/	1
	half	32,433	238	99.27%	87.12%	127	1.39
	all	32,234	37	99.89%	95.43%	563	3.38
sparc_exu	orig	76,709	1,006	98.69%	84.19%	/	1
	half	76,226	514	99.33%	91.76%	417	1.52
	all	75,745	39	99.95%	98.51%	1,745	2.68
sparc_lsu	orig	114,076	1,440	98.74%	83.20%	/	1
	half	113,427	787	99.31%	91.61%	649	1.18
	all	112,715	76	99.93%	98.18%	2,114	2.86
sparc_tlu	orig	104,694	1,553	98.52%	77.67%	/	1
	half	103,982	828	99.20%	87.53%	729	2.02
	all	103,224	70	99.93%	95.40%	2,165	3
sparc_ifu	orig	101,074	1,858	98.16%	66.93%	/	1
	half	100,351	1,123	98.88%	82.51%	820	1.28
	all	99,271	48	99.95%	91.29%	3,581	2.29
sparc_fpu	orig	157,728	1,159	99.27%	73.72%	/	1
	half	157,295	723	99.54%	88.53%	586	2.51
	all	156,639	63	99.96%	95.04%	2,376	4.04

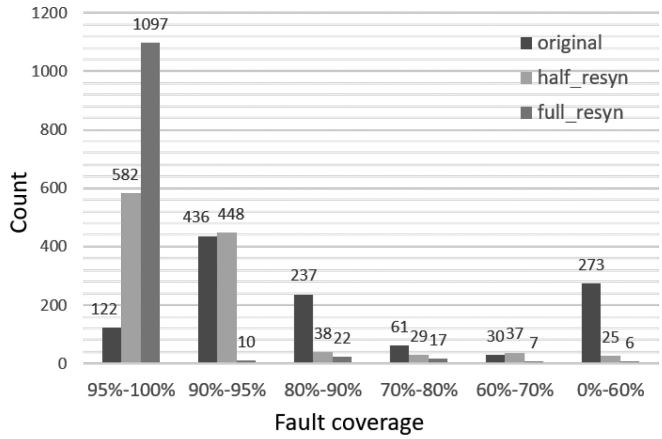

Fig. 5. Coverages of the original neighborhoods of undetectable faults in *sparc_fpu* after resynthesis.

(3) to reduce deterministic test pattern counts. We experimented with all the seven possible combinations of the three groups of test points. Of the seven designs of a circuit, we select the one with the highest coverage for the faults related to DFM guideline violations. We then apply the proposed layout resynthesis procedure to the selected design with p set to be 100%.

The results for several circuits are shown in Table 6. Under the name of the circuit, we show the number of nets in the circuit, followed by the number of inserted test points. For comparison,

Table 6. Circuits with Test Points

Circuit		F	U	Cov	Ave_c	Ntests	Nchanges	Rtime	Delay	Power	Area
b15 *(4722, 155)*	orig_ntp	21,103	713	96.62%	85.64%	378	/	1	100.00%	100.00%	100.00%
	resyn_ntp	20,408	10	99.95%	99.80%	378	2,398	7.02	98.27%	97.40%	100.00%
	orig_tp	21,960	184	99.16%	90.18%	256	/	1	108.34%	106.68%	104.16%
	resyn_tp	21,789	5	99.98%	99.25%	257	619	3.11	107.78%	104.93%	104.16%
DMA *(10696, 331)*	orig_ntp	44,589	409	99.08%	91.55%	996	/	1	100.00%	100.00%	100.00%
	resyn_ntp	44,191	2	99.99%	99.95%	997	821	6.63	99.63%	98.48%	100.00%
	orig_tp	46,303	286	99.38%	91.75%	650	/	1	100.70%	100.76%	102.98%
	resyn_tp	46,034	4	99.99%	99.89%	651	591	5.38	100.41%	99.32%	102.98%
sparc_ifu *(22257, 413)*	orig_ntp	101,074	1858	98.16%	66.93%	384	/	1	100.00%	100.00%	100.00%
	resyn_ntp	99,271	48	99.95%	91.29%	384	3,581	2.29	98.77%	99.13%	100.00%
	orig_tp	116,836	1539	98.68%	70.11%	201	/	1	105.48%	108.14%	102.82%
	resyn_tp	115,331	31	99.97%	93.45%	201	3,417	2.23	104.68%	107.46%	102.82%
sparc_fpu *(34418, 545)*	orig_ntp	157,728	1159	99.27%	73.72%	599	/	1	100.00%	100.00%	100.00%
	resyn_ntp	156,639	63	99.96%	95.04%	600	2,376	4.04	99.62%	98.83%	100.00%
	orig_tp	161,059	1073	99.33%	74.13%	232	/	1	103.78%	107.80%	101.79%
	resyn_tp	160,131	39	99.98%	95.78%	232	2,203	3.99	103.49%	107.13%	101.79%

in every case, we repeat the results for the circuit without test points in the first two rows. The last two rows describe the results for the circuit with test points. In both scenarios, we present the results of the original design in the first row and the results of the resynthesized design in the second row.

In addition to the columns defined for Tables 4 and 5, column *Ntests* in Table 6 provides the number of test patterns that detect all the detectable faults related to DFM guideline violations. In columns *Delay*, *Power*, and *Area*, we show the critical path delay, power consumption and die area of the circuit relative to the ones of the original design without test points. The increases in delay, power and area are due to the insertion of test points. The proposed layout resynthesis procedure does not allow any increases in delay, power, or area compared to the original design that it is applied to. It is possible to maintain the original design constraints by limiting the test points inserted to the circuit. However, in this article, we prefer not to interfere with the analysis and insertion of test points provided by the commercial tool.

It can be observed from Table 6 that the insertion of test points can improve the coverage of the circuit, while reducing the test pattern count. It also improves the coverage of the neighborhoods of undetectable faults. However, this coverage is still low in many cases. After applying the proposed layout resynthesis procedure, the coverage of the original neighborhoods of undetectable faults initially in the circuit increases significantly. Therefore, the application of the proposed procedure can also benefit the coverage for potential systematic defects when the circuit contains test points.

In some cases, column *Ave_c* of Table 6 shows that after applying the proposed procedure, the circuit with test points has an average coverage that is lower than the one of the circuit without test points. This is because the original layout of the circuit with test points is different from the original layout of the circuit without test points. The undetectable faults related to DFM guidelines in the two circuits can be different. Therefore, the unresolved undetectable faults in the two circuits due to design constraints can be different. In some cases, the coverage of the neighborhoods of certain unresolved faults in the circuit with test points may be low. As a result, the average coverage after applying the proposed procedure to the circuit with test points can be lower.

Table 7. Circuits after Logic Resynthesis

Circuit		F	U	Cov	Ave_c	Ntest	Nchanges	Rtime
b15	orig_inter	27,265	743	97.27%	83.19%	382	/	1
	resyn	26,533	11	99.96%	99.77%	382	2,461	7.23
DMA	orig_inter	53,567	413	99.23%	91.87%	1001	/	1
	resyn	53,156	1	99.99%	99.99%	1001	842	6.32
sparc_ifu	orig_inter	115,154	1,199	98.96%	73.15%	391	/	1
	resyn	113,988	32	99.97%	93.19%	391	2,910	2.01
sparc_fpu	orig_inter	185,598	1,517	99.18%	72.87%	609	/	1
	resyn	184,153	67	99.96%	95.18%	610	2,895	4.17

Table 8. Weighted DFM Guidelines

Circuit		F	U	Cov	Ave_c		Ntests	Nchanges	Rtime
					orig	resyn			
b15	spacing	21,092	702	96.67%	93.07%	93.54%	378	99	0.25
	via	20,973	582	97.23%	88.03%	93.51%	378	390	0.89
DMA	spacing	44,589	409	99.08%	93.12%	95.67%	996	31	0.19
	via	44,579	399	99.10%	92.27%	95.03%	996	104	0.77
sparc_ifu	spacing	100,998	1782	98.24%	78.63%	90.54%	384	601	0.48
	via	100,930	1713	98.30%	75.91%	90.18%	384	826	0.59
sparc_fpu	spacing	157,620	1051	99.33%	89.34%	96.23%	599	422	0.31
	via	157,563	994	99.37%	87.42%	95.82%	599	524	0.47

4.3 Circuits after Logic Resynthesis

In this section, we show the experimental results of applying the layout resynthesis procedure after applying the logic resynthesis procedure described in Reference [39]. This procedure addresses undetectable faults related to DFM guideline violations that are internal to cells by replacing cells with large numbers of undetectable internal faults by different cells. In this experiment, we focus on the undetectable external faults related to DFM guideline violations that remain in the circuit.

The results for several circuits are shown in Table 7. For every circuit, row *orig_inter* describes the original circuit after applying the logic resynthesis procedure in Reference [39]. Row *resyn* describes the circuit that is resynthesized using the layout resynthesis procedure described in this article. The columns of Table 7 are the same as in Tables 4, 5, and 6.

From Table 7, it can be observed that the circuits after logic resynthesis still contain large numbers of undetectable external faults related to DFM guideline violations, resulting in low-coverage areas in the design. This can be seen from columns *U* and *Ave_c*. The proposed layout resynthesis procedure addresses this issue by eliminating undetectable faults in the neighborhoods with low coverages. As a result, the coverage of the original neighborhoods of undetectable faults initially in the circuit increases significantly.

4.4 Weighted DFM Guidelines

In this section, we show the experimental results of applying the layout resynthesis procedure when different DFM guidelines have different weights.

We experimented with two cases for comparison. In the first case, the weights assigned to spacing related DFM guidelines are 50%. In the second case, the weights assigned to via dimension related DFM guidelines are 50%. We select the two types of DFM guidelines, since they are related

Table 9. Circuits with Higher Cell Utilization

Circuit		F	U	Cov	Ave_c	Nchanges	Rtime
b15	orig_70%	21,103	713	96.62%	85.64%	/	1
	resyn_70%	20,408	10	99.95%	99.80%	2,398	7.02
	orig_80%	21,489	811	96.23%	81.32%	/	1
	resyn_80%	20,702	14	99.93%	99.19%	2,491	8.19
	orig_90%	22,950	903	96.07%	82.06%	/	1
	resyn_90%	22,075	20	99.91%	99.31%	2,610	7.79
DMA	orig_70%	44,589	409	99.08%	91.55%	/	1
	resyn_70%	44,191	2	99.99%	99.95%	821	6.63
	orig_80%	45,033	524	98.84%	89.58%	/	1
	resyn_80%	44,530	7	99.98%	99.91%	1,037	7.18
	orig_90%	46,194	629	98.64%	88.37%	/	1
	resyn_90%	45,607	25	99.95%	99.36%	1,153	7.68
sparc_ifu	orig_70%	101,074	1858	98.16%	66.93%	/	1
	resyn_70%	99,271	48	99.95%	91.29%	3,581	2.29
	orig_80%	102,019	2105	97.94%	63.19%	/	1
	resyn_80%	100,029	97	99.90%	89.93%	3,710	2.96
	orig_90%	104,986	2973	97.17%	62.64%	/	1
	resyn_90%	102,173	156	99.85%	87.56%	4,002	3.59
sparc_fpu	orig_70%	157,728	1159	99.27%	73.72%	/	1
	resyn_70%	156,639	63	99.96%	95.04%	2,376	4.04
	orig_80%	163,579	1490	99.09%	74.56%	/	1
	resyn_80%	162,196	102	99.94%	91.63%	2,865	4.68
	orig_90%	170,479	1817	98.93%	72.10%	/	1
	resyn_90%	168,854	177	99.90%	90.29%	3,309	4.81

to most of the DFM guideline violations. In both cases, the weights assigned to the rest of the DFM guidelines are 5%, and p is set to be 80%.

The results for several circuits are shown in Table 8. For every circuit, row *spacing* describes the case when spacing related DFM guidelines are assigned higher weights, and row *via* describes the case when via dimension related DFM guidelines are assigned higher weights. The columns of Table 8 are the same as in Tables 4, 5, 6, and 7. In column *Ave_c*, we show the results for the original and the resynthesized circuits. In other columns, we show the result for the resynthesized circuit.

From Table 8, it can be seen that the number of changes to polygons in the first case is less than the one in the second case. This is because more DFM guideline violations that are translated to undetectable faults are related to via dimensions. This can also be validated from column *Ave_c*. When via dimension related DFM guidelines are assigned higher weights, the average coverage of the original neighborhoods of undetectable faults is typically lower than the one when spacing related DFM guidelines are assigned higher weights.

4.5 Circuits with High Cell Utilization

The cell utilization used thus far is 70% as discussed earlier. In this section, we show the results of applying the layout resynthesis procedure to circuits with higher cell utilization. We experimented with two cases for comparison. In the two cases, the cell utilization is set to be 80% and 90%, respectively.

The results for several circuits are shown in Table 9. The original layouts of all the circuits considered are obtained without design rule violations in all the cases. For every circuit, we repeat the results with 70% cell utilization, and then show the results with 80% and 90% cell utilization. In all the scenarios, we present the results of the original design in the first row and the results of the resynthesized design in the second row. The columns of Table 9 are the same as in Tables 4, 5, and 6.

It can be observed from Table 9 that the numbers of unresolved undetectable faults increase compared to the ones in the circuits with 70% cell utilization. This is due to the congestion in the circuit with high cell utilization. However, the proposed layout resynthesis procedure still reduces the number of undetectable faults significantly. Therefore, the coverage of the original neighborhoods of undetectable faults initially in the circuit increases significantly. This can be seen from column *Ave_c*.

5 CONCLUSIONS

We demonstrated that, among all the faults translated from DFM guideline violations, the undetectable external faults can result in areas of the circuit with poor coverage for potential systematic defects. Since the defects are likely to be systematic, the yield and DPPM can be impacted significantly. To address this issue, we defined a layout-based coverage metric and proposed a layout resynthesis procedure that makes fine changes to the layout while maintaining the design constraints. The proposed procedure is iterative. In every iteration, it prefers to eliminate the undetectable faults whose neighborhoods have the lowest coverages. The undetectable faults are eliminated by fixing the DFM guideline violations that lead to them. The experimental results for benchmark circuits and logic blocks of the OpenSparc T1 microprocessor showed that the coverages of the original neighborhoods of undetectable faults increase significantly. Therefore, the coverage for potential systematic defects can be improved significantly. We applied the procedure to circuits with test points. Experimental results showed that the coverage of the neighborhoods of undetectable faults is low even when circuits contain test points. After applying the proposed procedure, the coverage of the neighborhoods of undetectable faults increased significantly. A similar situation exists after logic resynthesis to eliminate undetectable faults that are internal to cells. We also experimented with weighted DFM guidelines and showed that the procedure is applicable when DFM guidelines have different levels of importance.

REFERENCES

[1] C. Acero, D. Feltham, F. Hapke, E. Moghaddam, N. Mukherjee, V. Neerkundar, M. Patyra, J. Rajski, J. Tyszer, and J. Zawada. 2015. Embedded deterministic test points for compact cell-aware tests. In *Proceedings of the International Test Conference*. 1–8.

[2] R. D. Blanton, F. Wang, C. Xue, P. K. Nag, Y. Xue, and X. Li. 2013. DREAMS: DFM rule evaluation using manufactured silicon. In *Proceedings of the International Conference on Computer-Aided Design*. 99–106.

[3] M. Brodsky, S. Halle, V. Jophlin-Gut, L. Liebmann, D. Samuels, G. Crispo, K. Nafisi, V. Ramani, and I. Peterson. 2005. Process-window sensitive full-chip inspection for design-tosilicon optimization in the sub-wavelength era. In *Proceedings of the IEEE/SEMI Conference and Workshop on Advanced Semiconductor Manufacturing*. 64–71.

[4] S. Chiu and C. A. Papachristou. 1991. A design for testability scheme with applications to data path synthesis. In *Proceedings of the Design Automation Conference*. 271–277.

[5] R. Desineni, L. Pastel, M. Kassab, M. F. Fayaz, and J. Lee. 2010. Identifying design systematics using learning-based diagnostic analysis. In *Proceedings of the Advanced Semiconductor Manufacturing Conference*. 317–321.

[6] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy, and I. Pomeranz. 2005. Methods for improving transition delay fault coverage using broadside tests. In *Proceedings of the International Test Conference*. 10 pp.–265.

[7] M. Fujita and A. Mishchenko. 2014. Efficient SAT-based ATPG techniques for all multiple stuck-at faults. In *Proceedings of the International Test Conference*. 1–10.

[8] C. Guardiani, N. Dragone, and P. McNamara. 2004. Proactive design for manufacturing (DFM) for nanometer SoC designs. In *Proceedings of the Custom Integrated Circuits Conference*. 309–316.

[9] P. Gupta, A. B. Kahng, I. Mandoiu, and P. Sharma. 2003. Layout-aware scan chain synthesis for improved path delay fault coverage. In *Proceedings of the International Conference on Computer-Aided Design*. 754–759.
[10] W. Howell, F. Hapke, E. Brazil, S. Venkataraman, R. Datta, A. Glowatz, W. Redemund, J. Schmerberg, A. Fast, and J. Rajski. 2018. DPPM reduction methods and new defect oriented test methods applied to advanced FinFET technologies. In *Proceedings of the International Test Conference*. 1–10.
[11] T. Jhaveri, A. Strojwas, L. Pileggi, and V. Rovner. 2008. Enabling technology scaling with "in production" lithography processes. In *Proceedings of the SPIE*, vol. 6924. 10.
[12] S. Kajihara, T. Sumioka, and K. Kinoshita. 1993. Test generation for multiple faults based on parallel vector pair analysis. In *Proceedings of the International Conference on Computer-Aided Design*. 436–439.
[13] D. Kim, M. E. Amyeen, S. Venkataraman, I. Pomeranz, S. Basumallick, and B. Landau. 2007. Testing for systematic defects based on DFM guidelines. In *Proceedings of the International Test Conference*. 1–10.
[14] D. Kim, I. Pomeranz, M. E. Amyeen, and S. Venkataraman. 2010. Defect diagnosis based on DFM guidelines. In *Proceedings of the VLSI Test Symposium*. 206–211.
[15] B. Krishnamurthy. 1987. A dynamic programming approach to the test point insertion problem. In *Proceedings of the Design Automation Conference*. 695–705.
[16] A. Krstic and K. Cheng. 1996. Resynthesis of combinational circuits for path count reduction and for path delay fault testability. In *Proceedings of the European Design and Test Conference*. 486–490.
[17] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede. 2004. Systematic defects in deep sub-micron technologies. In *Proceedings of the International Test Conference*. 290–299.
[18] S. Kundu and A. Sreedhar. 2011. Modeling manufacturing process variation for design and test. In *Proceedings of the Design, Automation and Test in Europe Conference*. 1–6.
[19] Y. Liu, E. Moghaddam, N. Mukherjee, S. M. Reddy, J. Rajski, and J. Tyszer. 2016. Minimal area test points for deterministic patterns. In *Proceedings of the International Test Conference*. 1–7.
[20] P. Maxwell, F. Hapke, M. RyynAd'nen, and P. Weseloh. 2017. Bridge over troubled waters: Critical area-based pattern generation. In *Proceedings of the European Test Symposium*. 1–6.
[21] E. Moghaddam, N. Mukherjee, J. Rajski, J. Tyszer, and J. Zawada. 2016. Test point insertion in hybrid test compression/LBIST architectures. In *Proceedings of the International Test Conference*. 1–10.
[22] A. Nardi and A. L. Sangiovanni-Vincentelli. 2004. Synthesis for manufacturability: A sanity check. In *Proceedings of the Design, Automation and Test in Europe Conference*. 796–801.
[23] OpenSPARC T1. 2006. Retrieved from http://www.oracle.com/technetwork/systems/opensparc/index.html.
[24] C. A. Papachristou, S. Chiu, and H. Harmanani. 1991. A data path synthesis method for self-testable designs. In *Proceedings of the Design Automation Conference*. 378–384.
[25] I. Pomeranz. 2014. Design-for-testability for multi-cycle broadside tests by holding of state variables. *ACM Trans. Des. Autom. Electron. Syst.* 19, 2 (2014), 19:1–19:20.
[26] I. Pomeranz. 2015. Enhanced test compaction for multicycle broadside tests by using state complementation. *ACM Trans. Des. Autom. Electron. Syst.* 21, 1 (2015), 13:1–13:20.
[27] I. Pomeranz and S. M. Reddy. 1995. On synthesis-for-testability of combinational logic circuits. In *Proceedings of the Design Automation Conference*. 126–132.
[28] I. Pomeranz and S. M. Reddy. 1999. On achieving complete coverage of delay faults in full scan circuits using locally available lines. In *Proceedings of the International Test Conference*. 923–931.
[29] I. Pomeranz and S. M. Reddy. 2010. On clustering of undetectable single stuck-at faults and test quality in full-scan circuits. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 29, 7 (2010), 1135–1140.
[30] I. Pomeranz and S. M. Reddy. 2010. On multiple bridging faults. In *Proceedings of the VLSI Test Symposium*. 221–226.
[31] S. Ravi and M. Joseph. 2014. High-level test synthesis: A survey from synthesis process flow perspective. *ACM Trans. Des. Autom. Electron. Syst.* 19, 4 (2014), 38:1–38:27.
[32] C. Schuermyer, K. Cota, R. Madge, and B. Benware. 2005. Identification of systematic yield limiters in complex ASICS through volume structural test fail data visualization and analysis. In *Proceedings of the International Test Conference*. 9–145.
[33] B. Seshadri, P. Gupta, Y. T. Lin, and B. Cory. 2012. Systematic defect screening in controlled experiments using volume diagnosis. In *Proceedings of the International Test Conference*. 1–7.
[34] A. Sinha, S. Pandey, A. Singhal, A. Sanyal, and A. Schmaltz. 2017. DFM-aware fault model and ATPG for intra-cell and inter-cell defects. In *Proceedings of the International Test Conference*. 1–10.
[35] J. E. Stine, J. Grad, I. Castellanos, J. Blank, V. Dave, M. Prakash, N. Iliev, and N. Jachimiec. 2005. A framework for high-level synthesis of system on chip designs. In *Proceedings of the International Conference on Microelectronic Systems Education*. 67–68.
[36] C. Tabery, M. Craig, G. Burbach, B. Wagner, S. McGowan, P. Etter, S. Roling, C. Haidinyak, and E. Ehrichs. 2006. Process window and device variations evaluation using array-based characterization circuits. In *Proceedings of the International Symposium on Quality Electronic Design*. 6 pp.–265.

[37] W. C. Tam and S. Blanton. 2011. To DFM or not to DFM? In *Proceedings of the Design Automation Conference*. 65–70.
[38] R. Turakhia, M. Ward, S. K. Goel, and B. Benware. 2009. Bridging DFM analysis and volume diagnostics for yield learning - A case study. In *Proceedings of the VLSI Test Symposium*. 167–172.
[39] N. Wang, I. Pomeranz, S. M. Reddy, A. Sinha, and S. Venkataraman. 2018. Resynthesis for avoiding undetectable faults based on design-for-manufacturability guidelines.
[40] S. Wang, Xiao Liu, and S. T. Chakradhar. 2004. Hybrid delay scan: A low hardware overhead scan-based delay test technique for high fault coverage and compact test sets. In *Proceedings of the Design, Automation and Test in Europe Conference*. 1296–1301.
[41] S. Wang, K. Peng, K. Hsiao, and K. S. Li. 2008. Layout-aware scan chain reorder for launch-off-shift transition test coverage. *ACM Trans. Des. Autom. Electron. Syst.* 13, 4 (2008), 64:1–64:16.
[42] S. Wang, K. Peng, and K. S. Li. 2006. Layout-aware scan chain reorder for skewed-load transition test coverage. In *Proceedings of the Asian Test Symposium*. 169–174.
[43] S. Wang and T. Yeh. 2007. High-level test synthesis for delay fault testability. In *Proceedings of the Design, Automation and Test in Europe Conference*. 1–6.
[44] J. Yang, N. A. Touba, and B. Nadeau-Dostie. 2012. Test point insertion with control points driven by existing functional flip-flops. *IEEE Trans. Comput.* 61, 10 (2012), 1473–1483.

Received October 2018; revised February 2019; accepted March 2019