Resynthesis for Avoiding Undetectable Faults Based on Design-for-Manufacturability Guidelines

Naixing Wang[1], Irith Pomeranz[1], Sudhakar M. Reddy[2], Arani Sinha[3], Srikanth Venkataraman[3]

[1] School of Electrical & Computer Eng., Purdue University, W. Lafayette, IN, USA
[2] Electrical & Computer Eng. Dept., University of Iowa, Iowa City, IA, USA
[3] Intel Corp., Hillsboro, OR, USA

Abstract— **As integrated circuit manufacturing advances, the occurrence of systematic defects is expected to be prominent. A methodology for predicting potential systematic defects based on design-for-manufacturability (DFM) guidelines was described earlier. In this paper we first report that, among the faults obtained based on DFM guidelines, there are undetectable faults, and these faults cluster in certain areas of the circuit. Because faults may not perfectly represent potential defect behaviors, defects may be detectable even though the faults that model them are undetectable. Clusters of undetectable faults thus leave areas in the circuit uncovered for potential systematic defects. As the potential defects are systematic, the test escapes can impact the DPPM significantly, and thus lead to circuit malfunction and/or reliability problems after deployment. To address this issue in the context of cell-based design, we propose a logic resynthesis procedure followed by physical design to eliminate large clusters of undetectable faults related to DFM guidelines. The resynthesized circuit maintains design constraints of critical path delay, power consumption and die area. The resynthesis procedure is applied to benchmark circuits and logic blocks of the OpenSPARC T1 microprocessor. Experimental results indicate that both the reduction in the numbers of undetectable faults and the reduction in the sizes of undetectable fault clusters are significant.**

I. Introduction

Aggressive scaling of integrated circuit (IC) technologies continues to decrease device size and increase circuit complexity. The continuous shrinking of device sizes increases the gap between the feature size and the lithography wavelength. As a result, certain layout features are more difficult to manufacture than others, and are more likely to lead to defects. Such features can cause repeated or systematic defects to occur when they are present multiple times [1-6]. Because of the systematic nature of these defects, they can impact the yield and defective-parts-per-million (DPPM) significantly and have a significant effect on the reliability of the design. To address systematic defects, appropriate design interventions are inevitable so as to remedy the potential manufacturing issues.

However, the constraints of die area, layout geometry and the ever-decreasing window of time to market make it impossible to obtain complete information about the potential manufacturing issues in advance. As a result, it is not possible to eliminate all the causes of systematic defects. Design-for-manufacturability (DFM) guidelines are recommended layout guidelines that attempt to capture and prevent layout features that may lead to yield and manufacturability issues. In contrast to design rules, which must be followed by the physical design process, DFM guidelines are applied when possible within the design constraints of area, delay and power for improving the yield. Since the number of DFM guideline violations can be very high, it is commonly the case that not all of them can be avoided. The relationship between DFM guideline violations and potential defects was first noted in [7]. In [7], layout locations where DFM guidelines are violated are found so as to anticipate the potential occurrence of systematic defects. DFM guidelines related to vias on interconnects and contacts on p-diffusion are considered in [7]. A more comprehensive list of DFM guidelines is considered in [8]. In recent work [9], DFM guidelines involving internal nodes of standard cells are considered. In all these works, the transistors affected by violations of DFM guidelines are identified at the schematic level, and the expected defect behaviors are translated to gate-level logic faults by using switch-level simulation. A target test set for these logic faults is then generated to close potential test holes and prevent adverse DPPM impact due to systematic defects. In [7], the target test sets consist of traditional test patterns. In [9], the translated gate-level logic faults are represented by input and output patterns of a cell. Such patterns define the so called user defined fault model (UDFM) for the potential defects they detect [11]. The UDFM is also used for modeling defects internal to standard cells in cell-aware testing [10-12]. The patterns generated at a cell boundary are translated to block-level test patterns and form the target test set.

Not all the faults that result from DFM guideline violations are detectable. When a translated logic fault is undetectable, it leaves an uncovered site in the circuit. This undetectable fault may also invalidate the tests for detectable faults nearby [13]. It was shown in [14] and [15] that undetectable faults in general tend to cluster in certain areas, leaving areas of the circuit uncovered. This can lead to test holes that affect more than a single gate or line. If an area of the circuit is uncovered, defects that can cause circuit malfunction and/or reliability problems after deployment may go undetected, resulting in test escapes. As we demonstrate in Section II, undetectable logic faults that result from DFM guideline violations also tend to cluster in certain areas of the circuit. Such faults may not perfectly represent every potential defect behavior. Thus, defects may be detectable even though the faults that model them are undetectable. Because these faults are used for modeling potential systematic defects, the test escapes caused by the clustering phenomenon can impact the DPPM significantly, and result in serious reliability problems.

A solution suggested in [14] and [15] is to target double faults that consist of an undetectable fault and an adjacent detectable fault. Additional tests for double faults were generated so as to improve the coverage of subcircuits containing undetectable faults. For the systematic defects considered in this paper, the coverage of the circuit needs to be even higher so as to avoid adverse DPPM impact. In addition,

The work of N. Wang and I. Pomeranz was supported in part by NSF Grant No. CCF-1714147.

978-3-9819263-2-3/DATE19/©2019 EDAA

experimental results show that the sizes of the clusters of undetectable faults resulting from DFM guideline violations are large. Thus, a significant number of additional test patterns is needed so as to achieve an acceptable coverage for theses defects. This may result in an excessive increase in the size of the test set, which leads to an unacceptable tester time.

Motivated by these observations, we propose a procedure that is based on logic resynthesis followed by physical design to eliminate or reduce large clusters of undetectable faults related to DFM guideline violations, and improve the coverage of the circuit for potential systematic defects. Logic resynthesis techniques are usually applied for optimizing the circuit with respect to delay, power and area [16-18]. They are also used for improving the testability of the circuit by reducing the difficulty of test generation [19-21]. More recently, logic synthesis-for-manufacturability is proposed [22-23]. It integrates manufacturability information into the traditional area-timing-driven logic synthesis, and uses a DFM extension library that contains yield-optimized cells so as to improve the manufacturability of the circuit. The procedure described in this paper is the first to address directly potential systematic defects that may remain uncovered and thus cause test escapes and lead to reliability issues. Specifically, the procedure targets clusters of potential systematic defects related to DFM guidelines when the clusters may remain uncovered. It leaves other areas unaffected by logic resynthesis. This is important for satisfying design constraints. In our experiments, the reduction in the sizes of undetectable fault clusters is significant, and typically one tenth of the original number of undetectable faults remain undetectable after applying the resynthesis procedure. This result cannot be obtained without considering faults directly.

The resynthesis procedure is developed in the context of a cell-based design. For this discussion we distinguish between faults that are internal to the standard cells (internal faults), and faults that are external to the standard cells (external faults). Every time a gate, or an instance of a standard cell, is used in the circuit, it introduces the same internal faults related to DFM guideline violations. The procedure eliminates the undetectable internal faults by resynthesizing the circuit with standard cells containing fewer internal faults. This is accomplished without changing the cell library, but only using different cells from the same library. Considering both internal and external faults, the total number of undetectable faults is only allowed to decrease monotonically when applying the resynthesis procedure. Therefore, the increase in the coverage of the circuit and the decrease in the sizes of large clusters of undetectable faults are significant when undetectable internal faults are eliminated by logic resynthesis.

The resynthesis procedure can be embedded into a standard cell based design flow. In a cell based design flow, a gate-level netlist and a layout are synthesized from an RTL description of a circuit using a standard cell library. Typically, several iterations of the design process are needed to satisfy design constraints such as area, delay and power. The proposed resynthesis procedure is also iterative, and it can fit within the overall iterative design flow. Specifically, an iteration of the design process can include one iteration of the resynthesis procedure to eliminate clusters of undetectable faults.

For a large chip, the proposed resynthesis procedure can be applied to every logic block separately so as to keep the computational effort acceptable. The proposed resynthesis procedure is applied to logic blocks of the OpenSPARC T1 microprocessor to demonstrate its applicability to such designs.

In this paper, the die area after applying the resynthesis procedure is kept the same as the original design so as to maintain the original floorplan of the chip. We denote by q the maximum acceptable percentage increase in delay and power compared with the original design. To obtain the smallest increase in delay and power that is needed for maximally reducing the clusters of undetectable faults, we start with $q = 0$, i.e., no increase in delay or power is allowed. After applying the resynthesis procedure, q is increased by one percent. We apply the resynthesis procedure with the increased q on top of the previous solution. We allow the maximum value of q to be increased up to five percent. Experimental results indicate that the coverage of the circuit can typically be improved significantly with the original floorplan and a small increase in delay or power. These results cannot be obtained by avoiding the larger cells with more internal faults altogether since these cells are needed for satisfying the design constraints. The resynthesis procedure can accommodate different design constraints if needed.

This paper is organized as follows. Section II demonstrates that undetectable faults related to DFM guidelines tend to cluster in certain areas of the circuit, and explains concepts that are relevant to this paper. Section III describes the details of the resynthesis procedure. Experimental results and analysis are presented in Section IV.

II. Clustering

We obtain a set of faults F by translating violations of DFM guidelines into likely shorts and opens inside and outside cells. We then translate the corresponding systematic defects into related stuck-at faults, transition faults, bridging faults and cell-aware faults modeled by UDFM [7-9].

A test generation procedure is applied to F. We denote by T a test set that detects all the detectable faults in F. The fault set $U = \{f_0, f_1, \ldots, f_{l-1}\}$ consists of all the undetectable faults in F.

We say that a gate *corresponds to a fault* f_i if (1) f_i is an internal fault, and it is inside the gate, or (2) f_i is an external fault, and it is on the inputs or outputs of the gate. An internal fault only has one gate that corresponds to it since it can only affect one gate. For an external fault, multiple gates may correspond to the fault when it is located on a net that connects multiple gates, or results in a short between two nets.

To explore the structural relations among the gates corresponding to the undetectable faults, we say that two gates are structurally adjacent if one of the two gates is directly driven by the other gate. For illustration, in Fig. 1, gates g1 and g2 are only adjacent in (c). We also define two faults f_a and f_b to be structurally adjacent if they are located on the same gate or two adjacent gates.

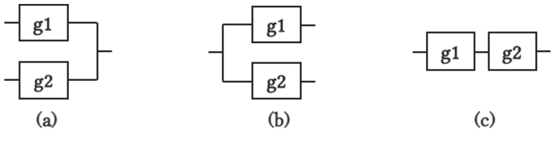

Fig. 1. Adjacent gates

We partition U into subsets S_0, S_1, … of adjacent faults. Initially, we set $S_i = \{f_i\}$ for $0 \le i < l$. For every pair of subsets of undetectable faults, S_{i1} and S_{i2} such that i2 > i1, we check whether S_{i1} and S_{i2} contain faults f_{i1} and f_{i2}, respectively, where f_{i1} and f_{i2} are structurally adjacent. If so, we merge S_{i1} and S_{i2} by adding the faults from S_{i1} to S_{i2}, and removing S_{i1}.

To demonstrate that the undetectable faults related to DFM guidelines and the gates corresponding to them tend to cluster in certain areas of the circuit, we computed the subsets of undetectable faults and the gates corresponding to them for benchmark circuits and logic blocks of the OpenSPARC T1 microprocessor. We denote the largest subset of adjacent undetectable faults by S_{max}. The set of gates corresponding to all the faults in S_{max} is denoted by G_{max}. The results are shown in Table I. Columns *F_In* and *F_Ex* show the numbers of internal and external faults related to DFM guidelines, respectively. Columns *U_In* and *U_Ex* show the numbers of undetectable internal and external faults related to DFM guidelines, respectively. Column *G_U* shows the number of gates that correspond to all the undetectable faults. Column *Gmax* shows the number of gates in G_{max}. Column *Smax* shows the number of undetectable faults in S_{max}. Column *%Smax_U* shows the percentage of all the undetectable faults that are in S_{max}.

From Table I, it can be observed that the circuits have large subsets of adjacent undetectable faults, and large sets of adjacent gates corresponding to them. Although the faults are undetectable, defects in the same areas of the circuit may be detectable, and they will go undetected if the areas are not covered. Figure 2 shows two large clusters of undetectable faults, cluster A and cluster B, as well as smaller clusters. A large cluster, such as cluster A or B, is missing tests for a large number of faults that are undetectable. The fact that tests are missing can allow detectable defects in the sites of cluster A or B to go undetected. The defects are detectable, even though the faults are not, because the faults may not model the defects perfectly.

In addition, despite the fact that the number of external faults related to DFM guidelines is larger than the number of internal faults, the major portion of the undetectable faults are internal faults. This is because the typical conditions required for detecting internal faults are stricter than the ones required for detecting external faults. Therefore, it is important to reduce the number of undetectable internal faults so as to improve the coverage of the circuit. Such faults can potentially be eliminated by replacing the gates corresponding to them. These observations motivate us to reduce the sizes of clusters of undetectable faults, and improve the fault coverage of the circuit, by resynthesizing subcircuits that consist of gates with undetectable faults.

TABLE I. CLUSTERED UNDETECTABLE FAULTS

Circuit	F _In	F _Ex	U _In	U _Ex	G_U	Gmax	Smax	%Smax _U
aes_core	15894	78364	5049	966	2705	911	1633	27.15%
des_perf	72654	281938	20209	688	5735	2638	10845	51.90%
sparc_exu	36791	79734	9747	1006	3661	2771	7072	65.77%
sparc_fpu	69979	164146	13381	1882	4685	2831	8291	54.32%

III. RESYNTHESIS PROCEDURE

In this section, we describe an iterative resynthesis procedure that eliminates undetectable faults so as to avoid clustering that can lead to poor coverage of certain areas of the circuit. The overview of the resynthesis procedure is given in Section III-A. In Sections III-B and III-C, we describe the details of the resynthesis procedure.

A. Overview

The circuit that is considered by the procedure is denoted by C_{all}. We assume that C_{all} was already optimized by one or more iterations of a standard IC design flow. The resynthesis procedure ensures that C_{all} improves in terms of the clustering of undetectable faults, while maintaining the design constraints of delay, power and area.

The proposed resynthesis procedure has two phases as shown in Fig. 2. The first phase focuses on the largest clusters of undetectable faults. In Fig. 2, the proposed procedure first targets *Cluster A* since it is the largest cluster. After several iterations, *Cluster A* is broken up by removing undetectable faults from it, and *Cluster B* becomes the current largest cluster. After breaking up *Cluster B*, all the large clusters in the circuit are eliminated. The second phase of the proposed procedure is then carried out so as to improve the coverage of the circuit further by removing undetectable faults from the entire circuit.

To guarantee that the proposed procedure maintains design constraints of critical path delay, power consumption and die area, we develop a backtracking procedure based on the observation that modifying fewer gates implies lower relative effect on design constraints. Details of the backtracking procedure will be discussed in Section III-C. In addition, the design overheads are sometimes reduced when gates with fewer internal faults, which are typically smaller, replace larger gates that can affect delay and performance adversely.

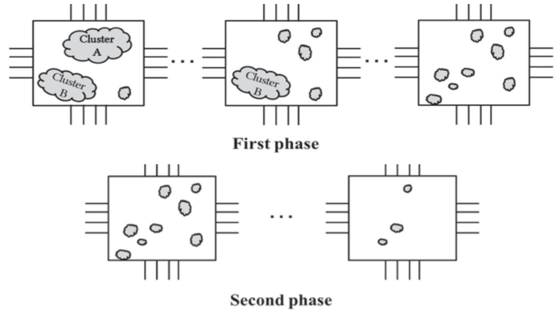

Fig. 2. Two phases of the proposed resynthesis procedure

B. Resynthesis with Two Phases

In this part, we describe the two phases of the resynthesis procedure.

The procedure considers different subcircuits of C_{all} in different iterations of every phase. The subcircuit as part of C_{all} that is considered for logic resynthesis in an arbitrary iteration is denoted by C_{sub}. In phase one, the procedure always targets the current largest cluster in the circuit, which is described by the set of faults S_{max} and the set of gates G_{max}. In this case, C_{sub} consists of all the gates in G_{max}. In phase two, C_{sub} consists of the gates that correspond to all the undetectable faults in the circuit.

C_{sub} is extracted using a commercial synthesis tool, and connected to other parts of C_{all} through the nets shared among them. The rest of the circuit, $C_{dont} = C_{all} - C_{sub}$, is not modified. We use G_{zero} to store the gates in C_{sub} that do not contain undetectable internal faults. The gates in G_{zero} are also not modified during logic resynthesis. This means that logic synthesis is applied only to $C_{\text{sub}} - G_{\text{zero}}$ during every iteration, and the rest of the circuit is not modified. This is important to avoid unnecessary design changes and computational effort.

We denote the logic synthesis tool used in this paper by *Synthesize*(). The physical design process that follows logic synthesis is denoted by *PDesign*(). The resynthesis procedure reduces the number of undetectable faults by eliminating undetectable internal faults. The internal faults are only related to the standard cells that are used in the circuit, and do not depend on the placement and routing processes of physical design. Therefore, *PDesign*() is called only when the number of undetectable internal faults decreases in the resynthesized circuit. This avoids unnecessary runtime for physical design. The resynthesis procedure calls *Synthesize*() and *PDesign*() iteratively as described next. The resynthesis repeats until the termination conditions described later in this section are satisfied.

For a subcircuit C_{sub}, the standard cells in the library are considered in the reverse order of the number of internal faults so as to eliminate undetectable internal faults. Let the library contain m standard cells, $cell_0$, $cell_1$, …, $cell_{\text{m-1}}$. A standard cell $cell_{\text{i}}$ is eligible to be considered by the procedure when (1) it is used to synthesize C_{sub}, (2) at least one gate in C_{sub}, of type $cell_{\text{i}}$, contains undetectable internal faults, and (3) $cell_{\text{i+1}}$, $cell_{\text{i+2}}$, …, $cell_{\text{m-1}}$ are sufficient for synthesizing C_{sub}. When $cell_{\text{i}}$ is considered, the procedure resynthesizes C_{sub} without using $cell_0$, $cell_1$, …, $cell_{\text{i}}$ to avoid introducing gates with more internal faults. The procedure then calls *Synthesize*() to resynthesize the subcircuit. A call to *PDesign*() is carried out if the number of undetectable internal faults decreases. Otherwise, the procedure moves on to consider the next standard cell. After calling *PDesign*(), if the acceptance criteria described later are satisfied, yet the resultant layout violates the design constraints, the backtracking procedure is invoked.

In both phases, a resynthesized circuit is considered for acceptance only if it satisfies the design constraints. The resynthesized circuit is accepted in phase one if the previous S_{max} reduces without increasing the total number of undetectable faults. We terminate phase one if the percentage of the faults in F that are in S_{max} reaches a target p_1. To balance the cluster sizes and the effectiveness of phase two, we experimented with different values of p_1. The results indicate that $p_1 = 1\%$ balances them well. In addition, we terminate phase one if S_{max} cannot be reduced further without increasing the total number of undetectable faults in the circuit.

We use p_2 to limit the size of S_{max} during phase two. p_2 is set to be the larger value between p_1 and the percentage of the faults in F that are in S_{max} after phase one. The resynthesized circuit is accepted during phase two if the total number of undetectable faults in the circuit decreases, while the percentage of faults in F that are in S_{max} does not exceed p_2. The second phase terminates when the number of undetectable faults in the circuit cannot be decreased further.

To speed up the procedure, additional conditions are used to terminate the current phase based on the following observation. In an arbitrary iteration, the procedure eliminates undetectable internal faults in C_{sub} by resynthesizing the circuit with cells that contain fewer internal faults. This can potentially increase the number of undetectable external faults, since the nets internal to the original gates may become external. Therefore, as the standard cells are considered, the gross trend of the number of undetectable faults in the circuit first goes down and then up. We terminate a phase when it appears that the number of undetectable faults is increasing.

C. Backtracking Procedure

In this part, we describe the backtracking procedure we use to guarantee that the resynthesized circuit does not violate the design constraints of delay, power and area. The backtracking procedure is called when an attempt to resynthesize the circuit fails because of the design constraints.

Suppose that the standard cell $cell_{\text{i}}$ is currently considered by the resynthesis procedure. Only $cell_{\text{i+1}}$, $cell_{\text{i+2}}$, …, $cell_{\text{m-1}}$ are allowed to be used for resynthesizing C_{sub}, since they contain fewer internal faults than $cell_{\text{i}}$. This indicates that all the gates in C_{sub} belonging to $cell_0$, $cell_1$, …, $cell_{\text{i}}$, which are not in G_{zero}, are considered to be replaced. These gates are included in a set G_{i}. Based on the observation that modifying fewer gates implies lower design overheads, the backtracking procedure considers subsets of G_{i} instead of trying to replace all the gates in G_{i}.

The procedure first removes gates from G_{i} in groups of $\sqrt{n}$ gates, where n is the initial number of gates in G_{i}. This is based on an analysis of the computational complexity, which is omitted for conciseness. The gates that are removed from G_{i} are placed in a set G_{back}. As the backtracking procedure removes more gates from G_{i}, higher numbers of undetectable faults are obtained. This may result in a resynthesized circuit that maintains the design constraints, yet does not satisfy the acceptance criteria. When this occurs, the backtracking procedure returns to G_{i} the last $\sqrt{n}$ gates that it added to G_{back} one by one.

With every modified G_{i}, the procedure calls *Synthesize*() to resynthesize C_{sub} without changing the gates in G_{back}, G_{zero} and C_{dont}. As before, *PDesign*() is called next when the number of undetectable internal faults decreases.

The resynthesized circuit is accepted if the acceptance criteria described in Section III-B are satisfied and no design constraints are violated. To speed up the resynthesis procedure, the backtracking procedure terminates whenever a resynthesized circuit is accepted. In addition, it terminates if no more gates can be added into or removed from G_{back}. If none of the resynthesized circuits is acceptable during backtracking, the current phase of the resynthesis procedure terminates.

IV. Experimental Results

The proposed procedure is applied to OpenCores® [24] benchmark circuits, and to logic blocks of the OpenSPARC T1 [25] microprocessor. OpenSPARC T1 is a 64-bit open-source microprocessor with eight cores, and each core can support up to four threads. Within OpenSPARC T1, we apply the proposed procedure to the logic blocks in a single SPARC core and the floating-point unit (fpu). We run the procedure on a Linux machine with 2.6GHz processors.

We obtained gate-level netlists and layouts from RTL descriptions using the tool kit in the standard cell library developed by OSU [26], which is based on TSMC 0.18um technology. This library contains 21 cells. The netlists obtained after logic synthesis for all the circuits considered in this paper are flattened. Each circuit is treated as one block with respect to floorplanning. The core utilization for the floorplan of the original physical design is set to be 70% for all the circuits.

As described in Section I, no increase in die area is allowed in this paper. In addition, the resynthesis procedure is applied to the circuit starting with $q = 0$, i.e., no increase in critical path delay or power consumption is allowed. q is allowed to increase up to five percent when applying the resynthesis procedure. It is important to note that the resultant circuit has been subjected to logic synthesis as well as physical design. Thus, the DFM related faults can be computed for the resynthesized circuit.

To define the set F of target faults, three categories of DFM guidelines are considered, *Via*, *Metal* and *Density*. The guidelines specify certain dimensions of width and spacing that are recommended for the physical design process. We use 19 guidelines in the *Via* category, 29 guidelines in the *Metal* category, and 11 guidelines in the *Density* category.

We use commercial tools for the logic synthesis and physical design processes. A commercial IC verification and sign-off package is used to find locations of potential systematic defects in the layout. In addition, we use a commercial automatic test pattern generation (ATPG) tool to generate test patterns for fault detection.

The experimental results are shown in Table II as follows. In each case, two rows correspond to a circuit. The first row describes the original design. The second row describes the resynthesized circuit obtained using the proposed resynthesis procedure with the largest value of $0 \leq q \leq 5$ that improves the coverage of the circuit. In row *average*, we show average values considering all the circuits before and after applying the resynthesis procedure.

In every row, column *Max Inc* shows the maximum acceptable increase in delay and power, q, or *orig* for the original design. Column F provides the number of faults modeling the potential systematic defects based on the DFM guidelines. Column U provides the total number of undetectable faults. Column *Cov* provides the coverage of the circuit, which is defined as *Cov* = 1-*U/F%*. Column T provides the number of tests required for achieving this coverage. Column *Smax* provides the number of undetectable faults in $S_{\max}$. Column *%Smax_all* provides the percentage of all the faults that are in $S_{\max}$. Column *Smax_I* provides the number of internal faults in $S_{\max}$. Column *%Smax_I* provides the percentage of all the faults in $S_{\max}$ that are internal faults.

In columns *Delay* and *Power*, we show the critical path delay and power consumption of the resynthesized circuit relative to the ones of the original design. In column *Rtime*, we show the run time for the proposed 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 guidelines. The test generation time is included since a test set is also obtained by the proposed resynthesis procedure.

From Table II it can be seen that the procedure described in this paper achieves a significant reduction in the number of undetectable faults for all the circuits considered. This can be seen from column U. With a small change to the total number of faults in the circuit, the coverage of DFM related faults increases significantly after applying the proposed procedure. From column T, it can be seen that the average number of tests required for achieving such coverage does not change much.

Because the resynthesis procedure focuses on the largest clusters in the first phase, and cluster sizes are also controlled in the second phase, the size of $S_{\max}$ decreases significantly. This can be seen from column *%Smax_all*. For most of the circuits, the percentage of all the faults that are in $S_{\max}$ after applying the proposed procedure is below 1%, which is the target value given by p_1.

TABLE II. EXPERIMENTAL RESULTS

Circuit	Max Inc	F	U	Cov	*T*	Smax	%Smax _all	Smax _I	%Smax _I	Delay	Power	Rtime
tv80	orig	29376	2677	90.89%	1445	1270	4.32%	938	73.86%	100%	100%	1
	0%	28908	465	98.39%	1493	381	1.32%	0	0%	93.61%	99.15%	19.10
systemcaes	orig	42360	4274	89.91%	778	2852	6.73%	2694	94.46%	100%	100%	1
	3%	40527	329	99.19%	804	192	0.47%	5	2.60%	96.21%	102.51%	29.17
aes_core	orig	94258	6015	93.62%	1217	1633	1.73%	1267	77.59%	100%	100%	1
	4%	97986	1691	98.27%	1287	281	0.28%	17	6.04%	96.21%	103.17%	18.68
wb_conmax	orig	193350	21334	88.97%	1211	5821	3.01%	5571	95.71%	100%	100%	1
	5%	183752	781	99.58%	1138	179	0.09%	0	0%	103.27%	104.43%	25.30
des_perf	orig	354562	20897	94.17%	518	10845	3.02%	10560	97.37%	100%	100%	1
	5%	362810	915	99.75%	498	59	0.02%	8	13.56%	104.91%	102.07%	17.21
sparc_spu	orig	41939	2598	93.81%	640	669	1.60%	656	98.06%	100%	100%	1
	3%	40584	296	99.27%	626	171	0.42%	5	2.92%	99.01%	102.18%	13.69
sparc_ffu	orig	48937	5155	89.47%	722	3554	7.26%	3232	90.94%	100%	100%	1
	1%	48721	629	98.71%	836	510	1.04%	17	3.33%	95.15%	100.29%	19.20
sparc_exu	orig	116525	10753	90.77%	1221	7072	6.07%	6338	89.62%	100%	100%	1
	3%	116562	770	99.34%	1292	688	0.59%	4	0.58%	96.19%	102.33%	19.21
sparc_ifu	orig	149116	10197	93.16%	1255	6619	4.44%	5513	83.29%	100%	100%	1
	0%	147376	1210	99.18%	1232	677	0.46%	7	1.03%	96.06%	99.54%	13.99
sparc_tlu	orig	151591	9603	93.67%	2622	5418	3.57%	4555	84.07%	100%	100%	1
	1%	151129	1036	99.31%	2740	740	0.49%	5	0.67%	92.11%	100.27%	17.14
sparc_lsu	orig	164658	9357	94.32%	925	5563	3.38%	4720	84.85%	100%	100%	1
	1%	161388	880	99.45%	934	578	0.36%	0	0%	100.16%	98.92%	15.53
sparc_fpu	orig	234125	15263	93.48%	1146	8291	3.54%	7515	90.64%	100%	100%	1
	0%	230597	3352	98.54%	1090	1998	0.86%	715	35.78%	94.89%	99.73%	16.37
average	orig	135066.42	9843.58	92.19%	1141.67	4967.25	4.06%	4463.25	88.37%	100%	100%	1
	resyn	134195.00	1029.50	99.08%	1164.17	537.83	0.53%	65.25	5.54%	97.32%	101.22%	18.72

The backtracking procedure helps ensure that the improvement in the coverage of the circuit is achieved under the design constraints. This can be observed from columns *Delay* and *Power*. In addition, the layouts for all the resynthesized circuits are achieved within the original floorplans without design rule violations. For some of the circuits, the delay or power decreases compared to the original design. This is because the proposed resynthesis procedure replaces certain larger gates with smaller gates that typically contain fewer internal faults. As a result, the adverse effects on delay and power that sometimes result from these larger gates are reduced.

It can also be observed that in general, the relative runtime does not increase as the complexity of the circuit increases. This is related to the fact that only subcircuits with gates corresponding to undetectable faults are resynthesized. In addition, physical design is carried out only when the number of undetectable internal faults decreases in the resynthesized circuit. As a result, the procedure described in this paper is applicable to complex logic blocks in large chips.

Finally, we investigated if the number and clusters of undetectable faults can be reduced by restricting the cells in the library to smaller gates with fewer internal faults. For illustration, we synthesized two of the circuits, *sparc_ifu* and *sparc-fpu,* when the seven cells with the largest numbers of internal faults were removed from the cell library used. We completed the layouts with the same floorplans as the original designs that use all the cells in the library. The critical path delays of *sparc_ifu* and *sparc-fpu* increased to 130% and 137%, respectively, and the power increased to 109% for both circuits, relative to the original design that uses all the cells in the library. This shows that simply avoiding the cells with the larger numbers of internal faults may not yield designs that maintain the design constraints for critical path delay and power, whereas the proposed procedure that resynthesizes specific circuit areas gives an effective method for eliminating undetectable fault clusters while maintaining design constraints.

V. Conclusion

We demonstrated that, among the faults that model potential systematic defects based on DFM guidelines, there are undetectable faults, and these faults cluster in certain areas of the circuit. This leaves areas in the circuit uncovered for systematic defects. The yield and DPPM can be impacted significantly by such systematic defects. To address this issue, we proposed a procedure based on logic resynthesis followed by physical design to eliminate large clusters of undetectable faults related to DFM guidelines, and thus improve the coverage of the circuit for potential systematic defects. The proposed resynthesis procedure has two phases. The first phase focuses on the largest clusters of undetectable faults. The second phase considers the entire circuit. A backtracking procedure was used to guarantee that the resynthesized circuit maintains design constraints of critical path delay, power consumption and die area. The experimental results for benchmark circuits and logic blocks of the OpenSparc T1 microprocessor showed that both the improvement in the coverage of the circuit and the reduction in the sizes of large clusters are significant.

References

[1] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede, "Systematic Defects in Deep Sub-Micron Technologies," in Proc. Int. Test Conf., 2004, pp. 290-299.

[2] C. Schuermyer, K. Cota, R. Madge and B. Benware, "Identification of systematic yield limiters in complex ASICS through volume structural test fail data visualization and analysis," in Proc. Int. Test Conf., 2005, paper 7.1.

[3] R. Turakhia, M. Ward, S. K. Goel and B. Benware, "Bridging DFM analysis and volume diagnostics for yield learning—A case study," in Proc. VLSI Test Symp., 2009, pp. 167-172.

[4] R. Desineni, L. Pastel, M. Kassab, M. F. Fayaz and J. Lee, "Identifying design systematics using learning based diagnostic analysis," in Proc. Adv. Semicond. Manuf. Conf., 2010, pp. 317-321.

[5] S. Kundu and A. Sreedhar, "Modeling manufacturing process variation for design and test," in Proc. Design Automation and Test in Europe Conf., 2011, pp. 1-6.

[6] B. Seshadri, P. Gupta, Y. T. Lin and B. Cory, "Systematic defect screening in controlled experiments using volume diagnosis," in Proc. Int. Test Conf., 2012, pp. 1-7.

[7] D. Kim, M. E. Amyeen, S. Venkataraman, I. Pomeranz, S. Basumallick and B. Landau, "Testing for systematic defects based on DFM guidelines," in Proc. Int. Test Conf., 2007, pp. 1-10.

[8] D. Kim, I. Pomeranz, M. E. Amyeen and S. Venkataraman, "Defect diagnosis based on DFM guidelines," in Proc. VLSI Test Symp., 2010, pp. 206-211.

[9] A. Sinha, S. Pandey, A. Singhal, A. Sanyal and A. Schmaltz, "DFM-aware fault model and ATPG for intra-cell and inter-cell defects," in Proc. Int. Test Conf., 2017, pp. 1-10.

[10] F. Hapke, R. Krenz-Baath, A. Glowatz, J. Schloffel, H. Hashempour, S. Eichenberger, C. Hora and D. Adolfsson, "Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs," in Proc. Int. Test Conf., 2009, pp. 1-10.

[11] F. Hapke, W. Redemund, A. Glowatz, J. Rajski, M. Reese, M. Hustava, M. Keim, J. Schloeffel and A. Fast, "Cell-Aware Test," IEEE Trans. on Computer-Aided Design, Sept. 2014, Vol. 33, No. 9, pp. 1396-1409.

[12] E. J. Marinissen, G. Vandling, S. K. Goel, F. Hapke, J. Rivers, N. Mittermaier and S. Bahl, "EDA solutions to new-defect detection in advanced process technologies," in Proc. Design Automation and Test in Europe Conf., 2012, pp. 123-128.

[13] X. Lin and J. Rajski, "The impacts of untestable defects on transition fault testing," in Proc. VLSI Test Symp., 2006, pp. 6-7.

[14] I. Pomeranz and S. M. Reddy, "On Clustering of Undetectable Single Stuck-At Faults and Test Quality in Full-Scan Circuits," IEEE Trans. on Computer-Aided Design, Jul. 2010, pp. 1135- 1140.

[15] I. Pomeranz, "On clustering of undetectable transition faults in standard-scan circuits," in Proc. VLSI Test Symp., 2011, pp. 128-133.

[16] P. Pan, "Performance-driven integration of retiming and resynthesis," in Proc. Design Autom. Conf., 1999, pp. 243-246.

[17] V. N. Kravets and K. A. Sakallah, "Resynthesis of multi-level circuits under tight constraints using symbolic optimization," in Proc. Int. Conf. Computer Aided Design, 2002, pp. 687-693.

[18] A. Saifhashemi, D. Hand, P. A. Beerel, W. Koven and H. Wang, "Performance and Area Optimization of a Bundled-Data Intel Processor through Resynthesis," in Proc. Int. Symp. Asynchron. Circuits Syst., 2014, pp. 110-111.

[19] S. Chiu and C. A. Papachristou, "A design for testability scheme with applications to data path synthesis," in Proc. Design Autom. Conf., 1991, pp. 271-277.

[20] A. Krstic and K.-T. Cheng, "Resynthesis of Combinational Circuits for Path Count Reduction and for Path Delay Fault Testability", in Proc. Europ. Design & Test Conf., 1996, pp. 486-490.

[21] S. Wang and T. Yeh, "High-Level Test Synthesis for Delay Fault Testability," in Proc. Design Automation and Test in Europe Conf., 2007, pp. 1-6.

[22] C. Guardiani, N. Dragone and P. McNamara, "Proactive Design for Manufacturing (DFM) for Nanometer SoC Designs," in Proc. Custom Int. Circ. Conf., 2004, pp. 309-316.

[23] A. Nardi and A. L. Sangiovanni-Vincentelli, "Synthesis for manufacturability: a sanity check," in Proc. Design Automation and Test in Europe Conf., 2004, pp. 796-801.

[24] http://www.opencores.org.

[25] http://www.oracle.com/technetwork/systems/opensparc/index.html.

[26] J. E. Stine, J. Grad, I. Castellanos, J. Blank, V. Dave, M. Prakash, N. Iliev, and N. Jachimiec, "A Framework for High-Level Synthesis of System-on-Chip Designs," in Proc. Int. Conf. Microelectronic Systems Education, 2005, pp. 67-68.