



# Increased Detection of Hard-to-Detect Stuck-at Faults during Scan Shift

Hui Jiang<sup>1</sup> · Fanchen Zhang<sup>2</sup> · Jennifer Dworak<sup>1</sup> · Kundan Nepal<sup>3</sup> · Theodore Manikas<sup>4</sup>

Received: 28 September 2022 / Accepted: 20 March 2023 / Published online: 25 April 2023  
© The Author(s) 2023

## Abstract

Test sets that target standard fault models may not always be sufficient for detecting all defects. To evaluate test sets for the detection of unmodeled defects,  $n$ -detect test sets (which detect all modeled faults at least  $n$  times) have previously been proposed. Unfortunately,  $n$ -detect test sets are often prohibitively long. In this paper, we investigate the ability of shadow flip-flops connected into a MISR (Multiple Input Signature Register) to detect stuck-at faults fortuitously multiple times during scan shift. We explore which flip-flops should be shadowed to increase the value of  $n$  for the least detected stuck-at faults for each circuit studied. We then identify which circuit characteristics are most important for determining the cost of the MISR needed to achieve high values of  $n$ . For example, circuits that contain a few flip-flops with upstream fault cones that cover a large percentage of all faults in the circuit can often achieve high  $n$ -detect coverage fortuitously with a low-cost MISR. This allows a DFT engineer to predict the viability of this MISR-based approach early in the design cycle.

**Keywords** DFT ·  $n$ -detect · Stuck-at faults · Scan shift capture · Defect coverage · MISR · Fault cone

## 1 Introduction

Abstract models of defects, known as faults, are generally targeted deterministically during test pattern generation. Historically, the most commonly used fault model has been

the stuck-at fault. With new defect mechanisms and more stringent quality requirements, the number of fault models has increased. More and more fault models, such as bridges, opens, transition faults, and path delay have appeared. While most of these fault models have focused on the interconnections between gates or standard cells in a design, more recently the cell-aware fault model [10–12, 16–19, 21, 31] has been proposed to better model the types of defects that may occur within standard cells. This model uses layout information from the standard cells to identify potential defect locations and analog circuit analysis to determine defect behavior. The detection conditions are then abstracted to logic values at the inputs to the standard cell that can be targeted with a traditional ATPG tool. For example, the authors of [11] successfully applied cell-aware test on an advanced library with their library characterization tool flow. Based on layout information, [12] optimized cell-aware test by indicating realistic locations of open defects and short defects. In a later work, [31] saved test time by not simulating defects at the analog level for a given pattern if that pattern cannot detect the defect at the logic level. In addition, several probabilistic approaches have been described, for example, pseudo-random and weighted random testing in [49]. Such approaches are commonly used for Logic Built-In Self Test (LBIST).

In addition to the probabilistic and deterministic approaches described above, a combination of these two methods is  $n$ -detect—where each modeled fault (e.g. stuck-at

---

Responsible Editor: P. Girard

✉ Hui Jiang  
huij@smu.edu

Fanchen Zhang  
fanzhan@cisco.com

Jennifer Dworak  
jdworak@mail.smu.edu

Kundan Nepal  
kundan.nepal@stthomas.edu

Theodore Manikas  
manikas@lyle.smu.edu

- <sup>1</sup> Electrical and Computer Engineering Department, Southern Methodist University, 6425 Boaz Lane, Dallas 75205, TX, USA
- <sup>2</sup> Cisco Systems, Inc., Cisco Building 16, 3700 Cisco Way, San Jose 95134, CA, USA
- <sup>3</sup> Electrical and Computer Engineering Department, University of St. Thomas, 2115 Summit Ave, St Paul 55105, MN, USA
- <sup>4</sup> Computer Science Department, Southern Methodist University, 6425 Boaz Lane, Dallas 75205, TX, USA

fault) is detected at least  $n$  times during test. The reasoning behind the  $n$ -detect approach arises from the realization that for the detection of any fault or defect, excitation (generating an error at the defect site) and observation (propagating the error to an observable output or scan cell) must be satisfied. The excitation requirements change from one type of fault or defect to another. However, the common requirement for detecting any fault or defect at a circuit site is that the value at that site must be observed [14]. Stuck-at faults are especially useful targets for  $n$ -detect because detecting all stuck-at faults at least  $n$  times guarantees that every circuit site is observed at least  $2n$  times:  $n$  times when the site value is one and  $n$  times when the site value is zero. Thus, targeting stuck-at faults multiple times increases the chance of the excitation requirements of an unknown and unmodeled defect being fortuitously met while the site is observed—hopefully leading to detection of the unmodeled defect.

Even though the methods mentioned above can increase fault coverage, all of these deterministic and probabilistic techniques may lead to large test volume, and correspondingly, long test time. The authors of [17] have previously shown that on average a 49% increase in pattern count was required when compared to a normal stuck-at test set when “cell-internal defects” and “traditional cell port faults” were targeted in ten industrial circuits. The same paper also showed that an average 2.5% increase in defect coverage was possible if a test set of the same length as a stuck-at test set was optimized for cell-aware fault detection. On the other hand, test sets generated for  $n$ -detect also require a significant number of additional patterns—especially for high values of  $n$ .

While many test compaction and compression approaches have been proposed to reduce test pattern counts and test data volume, alternative methods have enhanced Design for Testability (DFT) circuitry to try to make faults easier to detect. Common approaches include the use of test points to enhance the observability or controllability of lines in the circuit. For example, the authors of [34] used a method to capture test responses during scan shift that used test points to tap internal signals. They employed a technique that uses both observation scans and control points. Other approaches try to increase the flexibility of the flip-flops and scan chains in the design to control and capture data during test. For example, a test-per-clock approach was proposed in [35] that adds a multiplexer and XOR gate to every flip-flop in the design so that different chains may be in one of three modes during test: stimulus, compaction, or mission mode.

In [51], we proposed a method where a selected set of shadow flops was combined into a MISR (Multiple Input Signature Register) to detect cell-aware faults by capturing data during scan shift. Capturing data during shift allows a large increase in the number of effective testing clock cycles with no increase in the actual test time. Our experiments

showed that a significant fraction of the detectable static cell-aware faults that would otherwise be missed by a stuck-at test set could be captured during scan shift—reducing the number of additional patterns needed to achieve full cell-aware coverage.

As a bonus of the shift cycle capture approach we introduced in [51], in [24], we explored the ability of the same DFT structure to also increase the value of  $n$  for a stuck-at test set. In particular, we showed that, for some circuits, even with a very low percentage of flip-flops shadowed in the MISR, the detection counts of stuck-at faults that might have been detected only once with the original test set increased their detection count by at least a factor of ten. Such an increase could significantly improve the ability of test to detect untargeted faults and unmodeled defects.

In this paper, we aim to identify better flip-flop selection methods to improve the percentage of the originally low-detected faults that gain extra detections with the MISR. Larger MISRs that include more shadow flip-flops also require more routing and loading on internal circuit nodes. Thus, in addition to considering using the flip-flops selected for maximum cell-aware fault coverage in [51] in the MISR, we also propose three alternative flip-flop selection methods. Furthermore, our previous work has shown that the trade-off between flip-flop overhead and fault coverage can vary significantly from circuit to circuit. Thus, we also explore multiple ways of quickly determining whether or not a circuit is amenable to the proposed approach. Because the detection of the faults with the MISR relies on fortuitous detection, we explore the error introduced when testability is estimated using the random pattern simulation functionality in a commercial tool instead of simulating the actual “intermediate” patterns (i.e. the values that appear in the scan chain when the next ATPG pattern has only partially been shifted in). We will show why the easier-to-implement random pattern simulation serves as a useful approximation of the fault coverage that is likely to be achieved. Then, the ability of fault cone analysis to reveal fault testability and to indicate the hardware overhead needed for good fault coverage will be investigated. We will show that the fault cone analysis results further explain the different fault coverages and hardware overheads across circuits seen in this paper and in [24].

The rest of this paper is organized as follows. Section 2 describes some previous work in  $n$ -detect testing and fault cone analysis. Section 3 reviews the DFT modifications described in [51] for detecting static cell-aware faults and the experimental results we achieved in [24]. Section 4 describes the experimental setup of this paper, shows simulation results, and performs fault cone analysis to explain the simulation results we obtained, as well as the effect of circuit characteristics on the proposed scan chain enhancement. Section 5 concludes the paper.

## 2 Previous Work

The difficulty of ensuring that all potential defects are well-modeled and targeted has led many researchers to explore  $n$ -detect testing [1, 2, 5–9, 13, 14, 22, 25–28, 32, 33, 36–42, 47–49, 52].

Stuck-at faults have relatively simple excitation requirements. In particular, to excite a fault P stuck-at 0, a logic 1 must be placed at site P. In contrast, to excite fault P stuck-at 1, a logic 0 must be present at site P. Other types of static faults, such as bridges and cell-aware faults generally require the detection of a stuck-at fault to occur, with some extra conditions that need to be simultaneously satisfied. For example, in the case of a dominating bridge, the extra condition corresponds to the logic value at the other bridged site. In the case of a cell-aware fault, extra conditions may correspond to extra logic values that must be set at the inputs of a standard cell.

As a result, the test space (i.e. set of input patterns) that detects a static defect at a circuit site is often a subset of the patterns that detect a stuck-at fault at the same site. If multiple patterns are chosen randomly to detect the stuck-at fault multiple times ( $n$ -detect), the chance that at least one of them will also detect the untargeted faults or unmodeled defect increases. The more patterns that detect the stuck-at fault, the higher the probability is. This is the fundamental reason why higher values of  $n$  can improve defect detection.

Many researchers have investigated different aspects of  $n$ -detect test sets [1, 2, 5–9, 13, 14, 22, 25–28, 32, 33, 36–42, 47–49, 52]. The authors of [32] performed experiments on small physical circuits and found that no defects were missed when all stuck-at faults were detected at least 15 times and tests were applied at rated speed. The authors of [14] explained why  $n$ -detect test sets could be more effective due to the increasing number of site observations. The authors of [2] showed that the effectiveness of  $n$ -detect tests would increase if the layout were considered, and if the logic values in the physical neighborhood around a signal line were varied during additional detections. This was followed by experiments on physically-aware  $n$ -detect test sets applied to industrial circuits in [28]. The authors of [37] also investigated ways of varying the test patterns by using different propagation paths for different detections of the fault.

Other papers have looked at the value of adding one more detection for various values of  $n$  and showed that additional detections for small  $n$  are much more important than additional detections for large values of  $n$  [5, 6]. In fact, the value of additional detections decreases exponentially as the value of  $n$  increases [6]. Similarly, [42] looked at four definitions to measure the degree to which test quality began to saturate as the value of  $n$  increased. The generation of test sets that target the stuck-at faults detected rarely by a one-detect test set  $n$  times and target other faults until they are detected at

least  $m$  times ( $1 \leq m < n$ ) for detection of four-way bridging faults was studied in [38].

Analysis of test set sizes for  $n$ -detect test sets and ways of reducing those sizes were explored in [1, 8, 13, 22, 26, 27, 39–41, 52]. For example, [41] reordered the test patterns so that the first  $K$  patterns would increase unmodeled fault coverage (bridging faults and transition faults in the paper). This would allow the end of the pattern set to be truncated if necessary (e.g. when the entire set did not fit in the tester memory) while maximizing the quality of the remaining set.

An integer linear programming (ILP) algorithm for minimizing test set size was proposed in [26]. Later, in [22], another heuristic method was also applied to reduce the test set. In [39], for  $n$ -detection of transition faults, stuck-at fault patterns were reordered and selected to obtain  $n$ -detect transition coverage for the desired value of  $n$ . A test set generation method based on fault dominating relations was also proposed in [40] to reduce  $n$ -detect test size. Moreover, an Embedded Multi-Detect (EMD) ATPG Algorithm that utilizes the don't-care bits to achieve additional multiple detections of faults (stuck-at and bridging in the paper) without increasing the test set size beyond that of the original one-detect ATPG test set was described in [13]. Their approach is similar to the guaranteed and desired  $n$ -detect test set generation that has become available in some commercial software tools. However, because the number of don't care bits is reduced, the authors of [13] note that it will do less well in the presence of an on-chip decompressor. With extra hardware overhead, using logic implications for  $n$ -detect test set reduction was also investigated in [1]. However, [52] showed that ATPG tools trying to maximize the number of detections in a pattern could be biased against meeting some types of excitation conditions if those conditions were likely to prevent additional site observations.

Other papers have explored the use of weighted pseudo-random BIST for  $n$ -detect [49] and the relaxation of  $n$ -detect test patterns to increase the number of unspecified don't-care bits that can later be used for other applications such as low power or test compression [36]. The use of  $n$ -detect test sets for multiple-fault diagnosis was considered in [47, 48].

Meanwhile, fault cones have previously been studied in the context of multiple applications (e.g. [3, 9, 23, 29, 30, 37, 44, 46]). For example, based on fault cones, [37] proposed a new method to obtain various distinct propagation paths for faults for  $n$ -detect. The authors of [29, 30] used a fault cone technique to investigate internal structural information of the circuit to help identify failing scan cells for more accurate fault diagnosis. The authors of [44] introduced a method to test failures caused by replacing or cascading gates incorrectly in reversible circuits with fault cones. The researchers in [46] investigated extra constraints during test generation i.e. constraints for timing exceptions

to ensure the correctness of the pattern set) to improve test coverage decreased by timing exceptions, where fault cones helped in finding the hard timing exceptions. The authors of [23] graded the extent of strong fault-secureness with fault cone analysis (no fault-masking or path-sensitization if the fault cone sets are disjoint) for circuit robustness based on the detectability of faults. Similarly, also for robustness, in [3], an online fault-space pruning method for fault injection had been proposed. Analysis was performed by exploring the fault propagation to the next clock cycle state and fault masking with the circuit logic elements' fault cones.

This paper complements this previous work and will explore how our previously proposed DFT approach targeted toward cell-aware fault detection [51] can also improve the number of stuck-at fault detections with no increase in pattern count. In addition to the method applied in [24], the effect of four flip-flop selection methods and two types of patterns will be studied for the creation of the MISR in our DFT enhancement circuitry. Two of the flip-flop selection methods will use fault cones to reduce the time required to make the selection. Then, fault cone analysis will be further applied to study the circuit characteristics that influence whether a particular circuit is well-matched to our proposed approach [24]. This analysis will also apply to the results obtained in [50, 51] regarding the number of flip-flops that had to be shadowed in the MISR for five different circuits.

### 3 Detecting Defects Using a MISR Structure During Scan Shift

Unfortunately, much of the time and power expended during test are not directly used to detect defects. For example, a significant fraction of all testing clock cycles are generally not used for detecting defects when test sets are applied using scan chains. Instead, those clock cycles are devoted to shifting the next pattern into the chain while shifting out the previous pattern's capture data. Moreover, during scan shift, the circuit logic is generally not isolated from the chains, which generates a significant amount of toggling of the internal circuit sites during shift and leads to extra power draw during shift.

Previously, researchers have tried multiple approaches to try to reduce the clock cycles and power wasted during shift. This includes attempts to create multiple *shorter* chains (which may be fed by an on-chip decompressor) and thus reduces the shifting clock cycles (e.g EDT structure in [43] and Illinois Scan architecture in [4]). Similarly, the authors of [15] have applied this concept to JTAG by having multiple chains shifting in parallel. Adjacent fill and other low power shift approaches have also been proposed to reduce the test power [45].

However, even with the optimizations above that reduce the length of the chains, some shift cycles are still needed—and that time and power expended during shift are still “wasted” from the perspective of detecting new, as yet undetected, defects. Thus, in our previous work, we devised a DFT approach to detect defects during scan shift.

Essentially, the values in the chain in the middle of the shifting procedure each form an “intermediate pattern” that can be used for defect detection. However, to capture test results without overwriting the pattern being shifted in, some additional circuitry is needed to capture the results. Therefore, in [51], we used a MISR to capture data during scan shift for fortuitous cell-aware fault detection. High percentages (90% or more) of missed cell-aware faults were detected for all but one of the circuits we studied during scan-shift of stuck-at test patterns. The remaining circuit had almost 60% detection of the missed cell-aware faults during scan shift.

An example of this DFT-based approach is shown in Fig. 1 in the presence of an on-chip decompressor and compactor. While patterns are shifting into the scan chain, responses of the circuit are taken as inputs to a MISR, as shown in Fig. 1. We refer to the flip-flops in the MISR as shadow flops because their values are partially determined by the data that would normally be captured by the functional circuit flip-flops in normal mode, as opposed to shift mode. Of course, the data in each shadow flip-flop is also determined by the other values and feedback in the MISR.

Capturing data in the shadow flops that form the MISR instead of the original chain allows the original pattern/results to be kept, while results for intermediate patterns can be collected in the MISR. Moreover, the MISR makes it possible to obtain

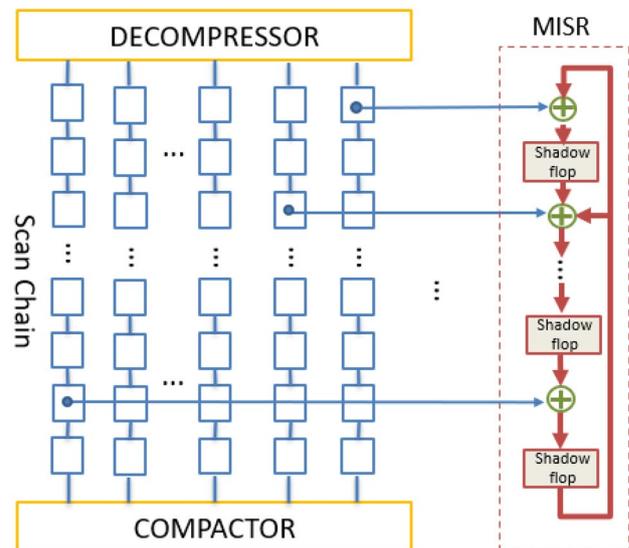


Fig. 1 Selected scan flip-flops are “shadowed” in a MISR so that a signature can be collected using intermediate shift data as test patterns

a single signature that combines test results over multiple scan shift cycles without saving the entire test data volume. Of course, good circuit simulation of the shift cycles that are used to capture test results must be performed and used to obtain the good circuit signature for the MISR. To save on simulation cost, it is possible to capture data only on a sample of all shift cycles, with a potential corresponding decrease in fortuitous fault coverage.

Given that intermediate shift stuck-at patterns are capable of detecting cell-aware faults fortuitously, it is reasonable to assume that these intermediate patterns can also detect other faults and unmodeled defects. In particular, previous work [24] has shown that the approach introduced in [51] can also significantly increase the number of times each stuck-at fault is detected, even for faults detected only a few times by a stuck-at test set. As previously discussed, the number of times a stuck-at fault is detected is important because each detection provides an additional chance of meeting the additional excitation requirements for detecting an unmodeled defect at a fault site above and beyond those needed for the detection of the stuck-at fault itself.

While [24] showed that the number of additional detections of stuck-at faults achievable with a MISR selected for the purpose of detecting cell-aware faults during scan shift was large, several questions remained to be answered. For example, [24] did not consider the possibility that even more detections could be achieved with a MISR designed to maximize the detections of stuck-at faults (instead of cell-aware faults) during shift. Moreover, the selection method was extremely time consuming and it required the explicit generation and simulation of the actual intermediate patterns—requiring a test set to be available as well. Thus, to explore better sets of flip-flops to be shadowed in the MISR and to estimate the percentages of improved stuck-at faults without long simulation time and intermediate pattern generation, three more flip-flop selection methods are studied in this paper. To save fault simulation time, fault cone analysis, which reports the flip-flops to which faults propagate, was used for this selection. To avoid the intermediate pattern generation procedure, random patterns were applied instead.

In addition, [24] showed that different circuits required significantly different overheads to achieve the desired additional detections. Fault cone data that explains the difference and an approach that could be used to predict whether a particular circuit was amenable to a low-cost MISR were not presented. Thus, the following sections will also present experimental results that explore these questions.

## 4 Experimental Procedure and Results

Experiments in this paper were run on five circuits obtained from Opencores.org [20]. Table 1 shows the characteristics of each circuit (number of flip-flops in column

2, number of stuck-at faults in column 3, number of stuck-at ATPG patterns in column 4, and number of intermediate patterns in column 5).

The overall procedure for data collection in this paper is shown in Fig. 2. Applying the procedure with stuck-at ATPG patterns and intermediate patterns will be illustrated first as an example. Then, the procedure with random patterns will be introduced.

First, stuck-at ATPG patterns were obtained using a commercial tool, and intermediate patterns were extracted using methods introduced in [51]. Each intermediate pattern is a combination of part of a new shift-in pattern and data captured from the circuit during the previous test. For example, a scan chain with  $m$  flip-flops corresponds to  $m - 1$  intermediate shift patterns for each pattern shifted into the circuit under test (CUT). The number of patterns simulated for each circuit is shown in Table 1.

Then, to select the set of flip-flops to be shadowed in the MISR in an efficient and effective manner, ordered lists of flip-flops were generated with four methods. All four methods for flip-flop selection are listed in Table 2. For each method, flip-flops were sorted in the order of the number of faults covered by each flip-flop in the circuit from maximum to minimum. Method 1 will be used as an example to explain the experimental procedure. Then, simulation results with all four flip-flop selection methods and two types of patterns will be shown. More details of Method 2, 3 and 4, and the other type of pattern will be explained later.

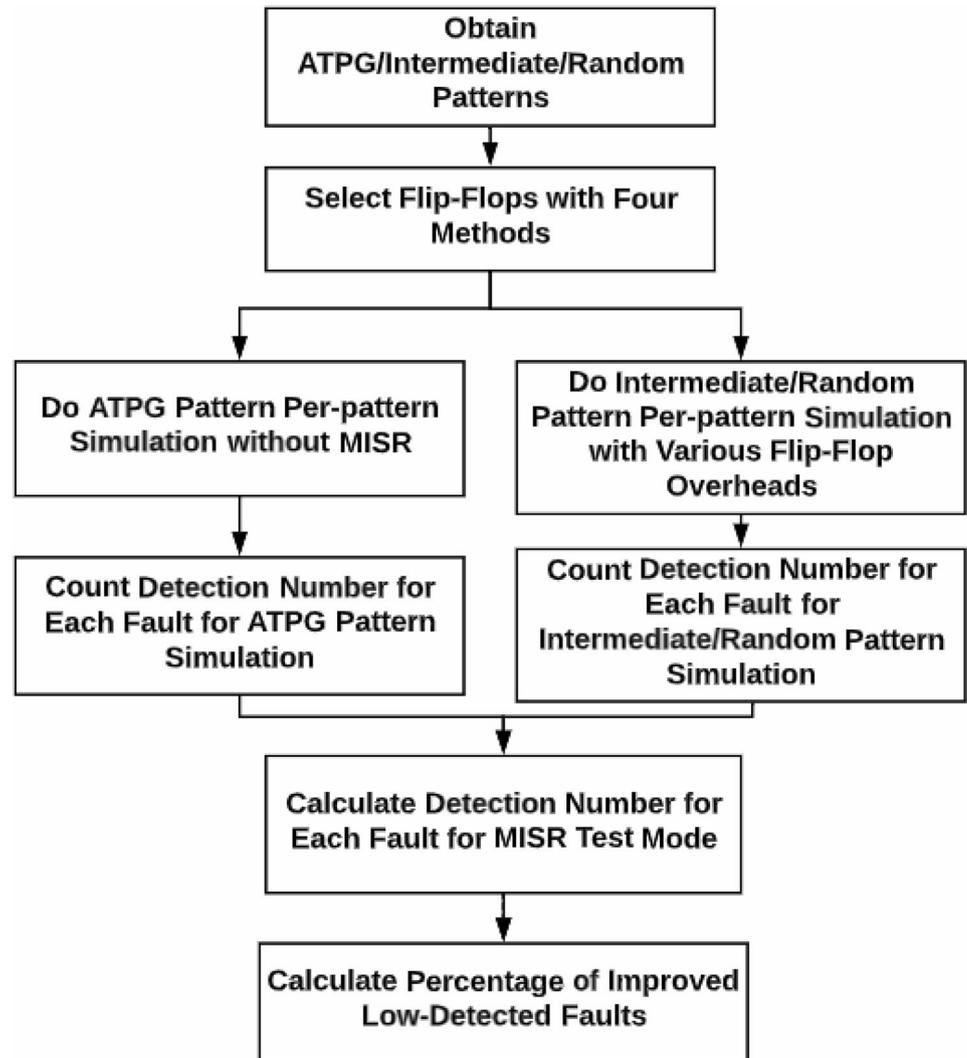
### 4.1 Flip-flop Selection for Maximum Cell-Aware Fault Coverage (Method 1) and Experimental Procedure

Method 1 corresponds to the flip-flop selection method introduced in [51]. With Method 1, flip-flops were selected for maximum coverage of detectable cell-aware faults that would not be fortuitously detected by the stuck-at ATPG test patterns applied in the normal way. The following greedy algorithm was used to select the flip-flops. First, through fault simulation of stuck-at ATPG patterns and the corresponding intermediate patterns, lists of cell-aware faults detectable at each flip-flop were obtained. Then, the

**Table 1** Characteristics of Circuits

Circuit	# of Flip-Flops	# of Stuck -At Faults	# of ATPG Patterns	# of Intermediate Patterns
Quadratic	120	8166	36	1044
Des56	193	13788	119	2856
Fm_rec	501	19888	406	10962
Colorconv	584	38518	98	3136
Fpu	5231	297358	538	17216

Fig. 2 Experimental Flowchart



flip-flops were sorted by the number of cell-aware faults they could detect from maximum to minimum. The flip-flop at the top of the list with the highest cell-aware fault detections would be selected to be shadowed in the MISR.

After that, faults covered by the selected flip-flop would be removed from the fault list. The remaining flip-flops would be resorted in the order of number of cell-aware faults they could detect that were still included in the fault list. The

**Table 2** Methods for Selecting Flip-Flop Lists to Be Shadowed in the MISR

Method Reporting	Fault List Selection	Flip-Flop	Full Fault List
1. Selected for Max Cell-Aware Fault Coverage	Simulated with all stuck-at ATPG and intermediate patterns	Greedy algorithm	Cell-aware faults that can be detected by cell-aware ATPG patterns but missed by stuck-at ATPG patterns
2. Selected for Max Stuck-At Fault Coverage	Simulated with all stuck-at ATPG and intermediate patterns	Greedy algorithm	Stuck-at faults that can be detected by stuck-at ATPG patterns
3. Selected from Fault Cone Report without Considering Overlap	Fault cone report	Sorting based on Uncontrolled (UC) faults	N/A
4. Selected from Fault Cone Report Considering Overlap	Fault cone report	Greedy algorithm	All unique UC faults reported to D pins of all FFs

process would continue until the desired coverage was achieved or all flip-flops are selected. If the fault list became empty before the required number of flip-flops were selected, then the remaining flip-flops were selected in order from the last available sorted list. Thus, an upper bound for hardware overhead occurs when 100% of the flip-flops are shadowed.

Although Method 1 selects the flip-flops based upon cell-aware fault detections, we are still interested in the ability of the MISR to detect stuck-at faults multiple times. As a result, all stuck-at ATPG patterns generated by the commercial tool were simulated with the entire scan chain being used for fault detection. This provided the normal fault detection number for all of the stuck-at faults. Then, the intermediate patterns (no stuck-at ATPG patterns included) were also simulated with flip-flops shadowed in the MISR, while all flip-flops that would not be included in the MISR were masked. This ensured that only the shadowed flops could be used for fault detection. The sum of these two detection numbers for each fault corresponds to the total detection number achieved during test using scan shift with the MISR. The experiments were run with the same hardware overheads for all circuits, i.e. 1%, 2%, 4%, 8%, 16%, 32% and 100% (the number of flip-flops to be shadowed was rounded up). The number of extra detections of each of these faults when the MISR was used was calculated. Then, the percentage of the hard-to-detect faults whose detection counts increased was calculated for each MISR hardware overhead.

It has previously been shown that the circuit locations with the lowest stuck-at fault detection numbers are most likely to contribute to test escapes (e.g [1, 8, 9, 27, 32, 33, 40]). Thus, we focused on faults that would be detected only 1 to 15 times ([8, 32]) without the use of MISR. The number of extra detections of each these faults when the MISR was used was calculated. Then, the percentage of the hard-to-detect faults whose detection counts increased was calculated for each MISR hardware overhead.

Figure 3 shows simulation results for the five different circuits. Each sub-figure (a), (b), (c), (d), and (e) of Fig. 3 corresponds to a different circuit. The x-axis corresponds to the hardware overhead of the MISR, and the different bars correspond to the different methods of selecting and analysing the flip-flops to be included in the MISR. The data for Method 1 as described above corresponds to the first bar on the left of each overhead cluster.

As can be seen in Fig. 3, with Method 1, at least some of the low-detected faults are detected more times for all of the hardware overheads. Obviously, as the hardware overhead increases, the number of faults whose detection count increases goes up as well. However, even with only 1% hardware overhead, the number of faults whose detection counts increase is usually significant.

Having a MISR can give rise to aliasing, which can lead to faults in the circuit being masked when the faulty

signature is equal to the correct signature. Note that in this paper, we have not quantified the degree to which aliasing in the MISR would lead to some missed stuck-at fault detections. However, experiments performed by our group in [50] showed very low aliasing for cell-aware faults for those simulated MISRs.

As shown in Table 3, with the flip-flops selected for maximum cell-aware fault coverage (column 2), 0 to 4 cell-aware faults were missed (column 4), which were negligible given that thousands of faults were present in the circuits (column 3). We do not anticipate that aliasing would lead to a significant reduction in stuck-at fault detections either unless the MISRs become very small.

#### 4.2 Flip-flop Selection for Maximum Stuck-At Fault Coverage (Method 2)

Method 1 selected the flip-flops based on cell-aware fault coverage. In Method 2, we investigated the effect of targeting the stuck-at faults instead. Thus, similar to Method 1, Method 2 used the greedy algorithm to iteratively select flip-flops. However, the selection was based on stuck-at fault coverage. Because the goal was to increase the number of times that each stuck-at was detected, the full stuck-at fault list was targeted with this method.

The corresponding results for Method 2 with intermediate patterns correspond to bar 3 of each hardware overhead cluster in Fig. 3. The figure shows that the percentage of low-detected stuck-at faults detected with the shadow flops is generally higher than Method 1 when stuck-at faults are targeted in the flip-flop selection method when at least 2% hardware overhead is used. However, this is not always true for a hardware overhead of 1%. This is likely due to the fact that *all* stuck-at faults were targeted during flip-flop selection. Because Method 1 targeted cell-aware faults not detected by the ATPG test set, the targeted cell-aware faults were more likely to be present at hard-to-observe circuit locations. These correspond to the same locations that are likely to have few stuck-at fault detections. Thus when the allowed overhead of the MISR is small, it is important to target the least detected stuck-at faults if the goal is to achieve better stuck-at  $n$ -detect fault coverage of the least detected faults.

To show more detail of how Method 2 can improve the number of detections of the low-detected stuck-at faults, average detection numbers are shown in Fig. 4. Faults are grouped by the original detection number without the MISR. Thus, the x-axis corresponds to the original detection number (1-15) with the stuck-at ATPG patterns. A smaller original detection number means that the extra detections that can be achieved with the MISR are more important. In fact, in [6], it was shown that the value of an *additional* observation of a circuit site with few observations (and therefore

few stuck-at fault detections) decreases exponentially as the number of current observations increases. Thus, for example, going from 14 to 15 observations is less valuable than going from 1 to 2 observations, but more valuable than going from 100 to 101.

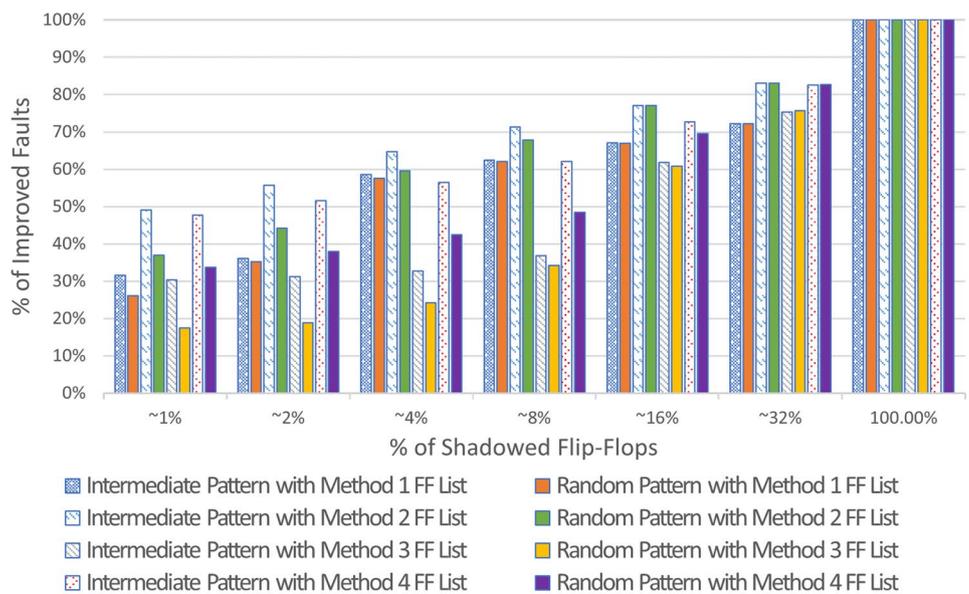
On the y-axis, average detection numbers are shown after inserting the MISR with flip-flop selection overheads that correspond to the overheads from Fig. 3. The y-axis is log scaled to better show the increases corresponding to the smallest detection numbers. For most of the circuits, even the lowest original detection group (i.e. 1 detection) gains at least one and often significantly more than

one additional detection on average even for 1% flip-flop overhead. The lowest average detection number for the 1% flip-flop overhead MISR appears in Colorconv and is 1.8 detections on average.

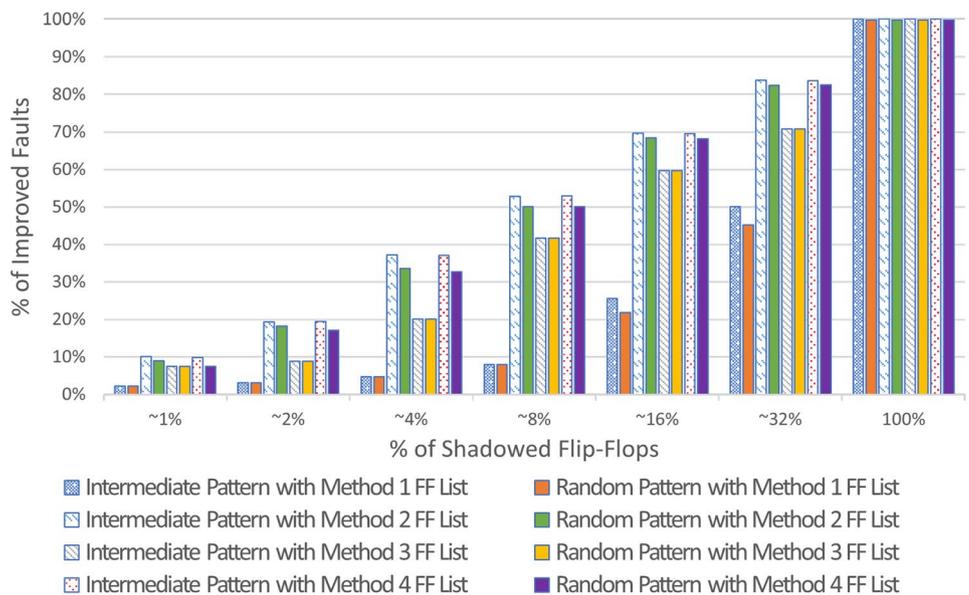
### 4.3 Flip-flop Selection from Fault Cone Results (Methods 3 and 4)

Even though Methods 1 and 2 can improve the detection number of low-detected stuck-at faults, the simulation time required for these two methods is considerable. This is partly due to the limitations of the software tool we used, which

**Fig. 3** Percent of Low-Detected Faults that Gained Extra Detections with the MISR

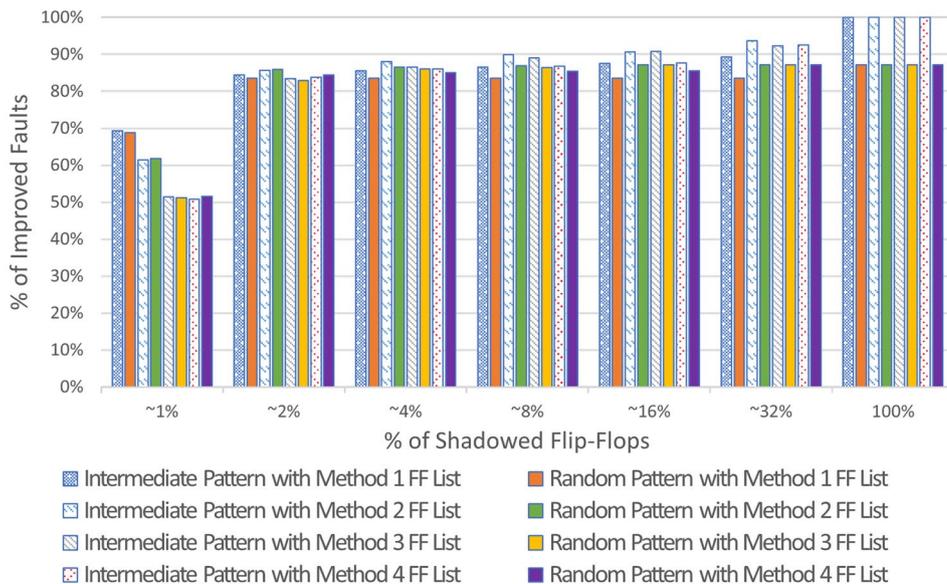


**(a) Quadratic**

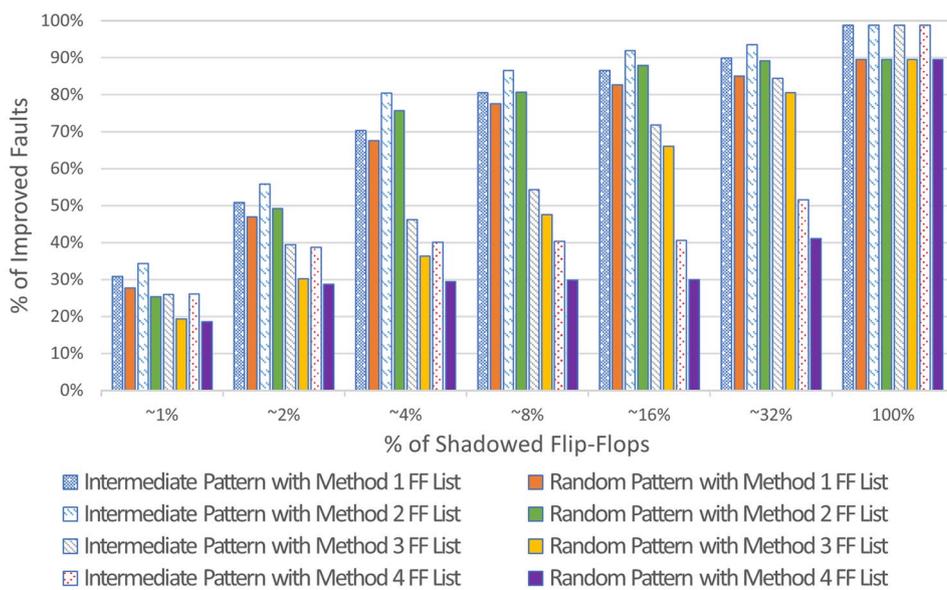


**(b) Des56**

Fig. 3 (continued)

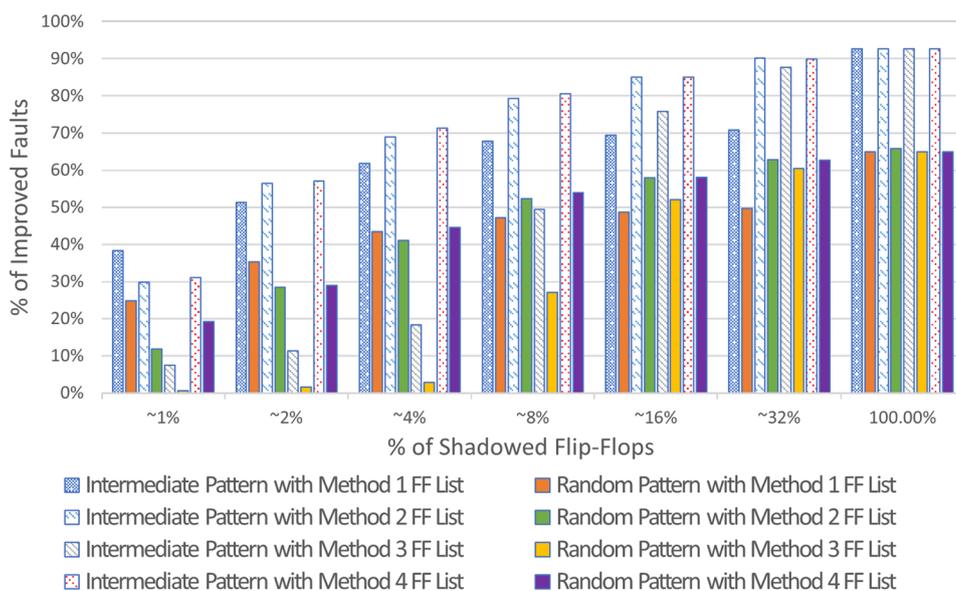


(c) *Fm\_rec*



(d) *Colorconv*

Fig. 3 (continued)



(e) Fpu

required the simulations of the intermediate patterns to be repeated for each flip-flop to determine which faults were detected at the individual flip-flops for flip-flop selection. (If internal support were provided by the software to obtain this data directly in a single simulation, it would be much quicker.) To quickly estimate the effect of the MISR without spending as much simulation time as was required with Methods 1 and 2, fault cone methods (Method 3 and Method 4) are proposed. For example, for the largest circuit that we tested (circuit Fpu), Methods 1 and 2 required a few days of per flip-flop simulation using the software we had available, while the fault cone report only required a few minutes.

Fault cone methods generate detectable fault lists based on circuit structure. Faults reported to the D pin of a certain flip-flop means that the D pin of this flip-flop is in the downstream cone of influence of the faults reported. In other words, it may be possible for these faults to be propagated to this flip-flop and to be captured in this flip-flop during test or simulation. Thus, fault cone results show the possibility of flip-flops detecting faults. According to our fault cone report of the five experimental circuits ([20]),

Table 3 Cell-aware faults missed due to MISR aliasing

Circuit	Flip-flops in MISR	# of Cell-Aware Faults	# of Missed Cell-Aware Faults
Quadratic	34	~34500	0
Des56	68	~54000	0
Fm_rec	11	~79000	2
Colorconv	38	~184000	0
Fpu	269	~1210000	4

six fault classes were reported, i.e. ATPG\_Untestable (AU), DET\_Implication (DI), Tied (TI), UNControlled (UC), UNUsed (UU), Blocked (BL) faults. Based on the definition of these fault classes, we considered only UC faults to be potentially detected with pattern stimulus. Thus, we considered the UC fault number as the detection metric for the fault cone method.

Method 3 sorted flip-flops based on the number of UC faults reported to each flip-flop, regardless of overlap among fault lists of flip-flops. In contrast, Method 4 sorted flip-flops according to number of uncovered UC faults in the full UC fault list of the circuit from maximum to minimum and used the greedy algorithm from Methods 1 and 2 for selection. This meant the faults already covered by selected flip-flops would be removed from the target UC fault list.

Although the fault cone method was employed to avoid simulation while selecting the flip-flops for Methods 3 and 4, to determine the real effect of these methods, simulation was done to evaluate their effectiveness. Results with intermediate patterns are shown as bar 5 of each hardware overhead sample cluster in Fig. 3 for Method 3, and bar 7 for Method 4. Both Methods 3 and 4 yield lower values for the percentage of improved faults than Method 2. However, the fault cone method is much quicker. Because it uses the greedy algorithm that considers which stuck-at faults have already been detected and removes them from the fault list, the results are better for Method 4 than for Method 3 for most of the circuits (i.e. except for the circuit Colorconv). The results for Colorconv may be due to the fact that fault cone analysis can be a less accurate way to estimate fault detections versus simulating with a pattern stimulus.

### 4.4 Random Patterns Simulation

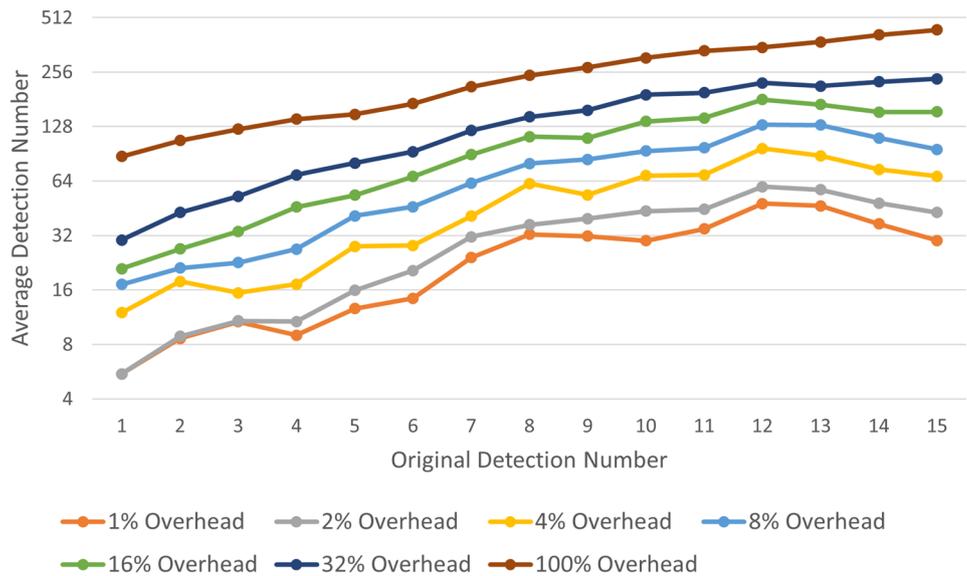
While fault cone flip-flop methods (Method 3 and 4) did not require simulation time when selecting flip-flops, our analysis still used simulation of intermediate patterns from the actual test set to determine their effectiveness for improving detection counts. To obtain such an analysis without the need to already have an ATPG test set and generate the intermediate patterns, here we consider the simulation of random patterns instead.

The same number of random patterns as intermediate patterns were simulated with the same flip-flops in the MISR for each method, and the results were processed with the

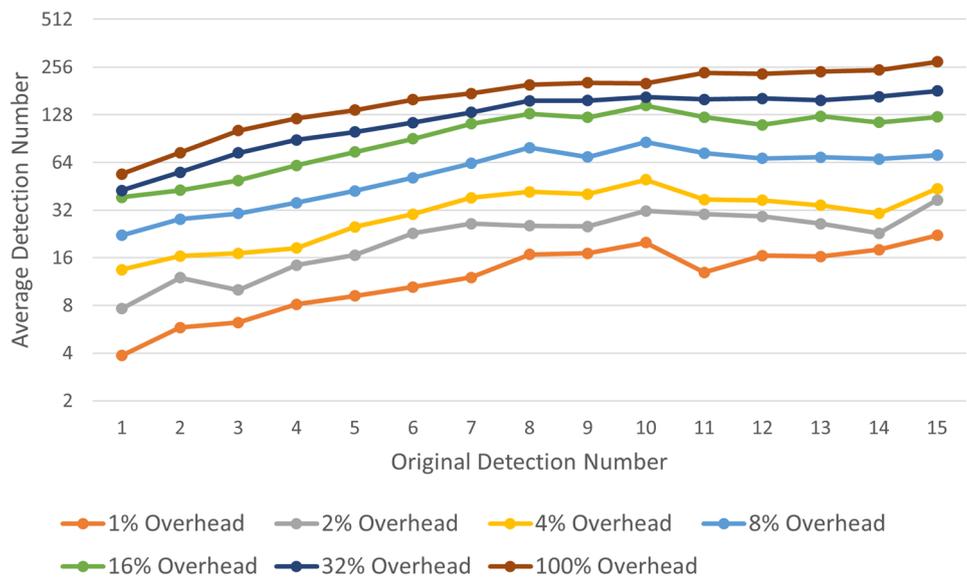
same procedure as described above. Simulation results in bars 6 and 8 of each flip-flop sample cluster show the results with random patterns. The percentage of improved faults are slightly lower than the simulation results obtained with intermediate patterns (bar 5 and 7), except for Fpu (where larger differences are shown). Thus, with fault cone selection methods (Method 3 and 4) and random patterns, the user can quickly estimate the effect of this MISR method.

The flip-flops selected with Methods 1 and 2 were also simulated with random patterns to further compare the effect of simulating intermediate versus random patterns. Each simulation result with random patterns can be seen in bars 2 and 4 of each flip-flop sample cluster in Fig. 3. When

**Fig. 4** Average Detection Number for Low-Detected Faults with Flip-Flops Selected for Stuck-at Faults

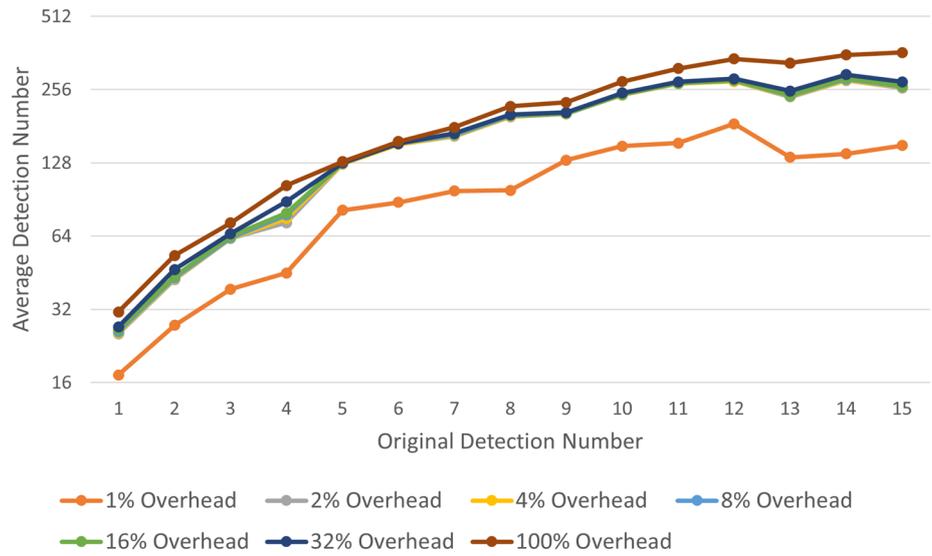


**(a) Quadratic**

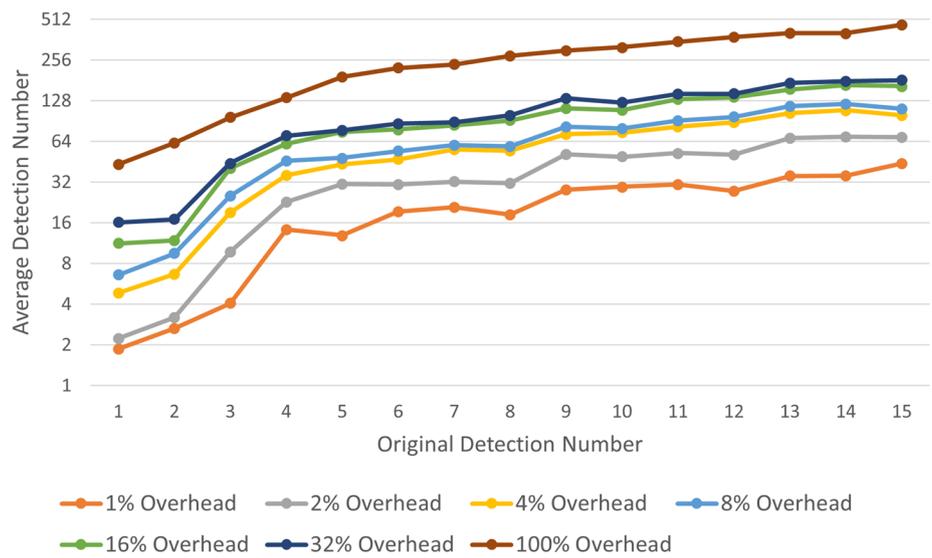


**(b) Des56**

Fig. 4 (continued)

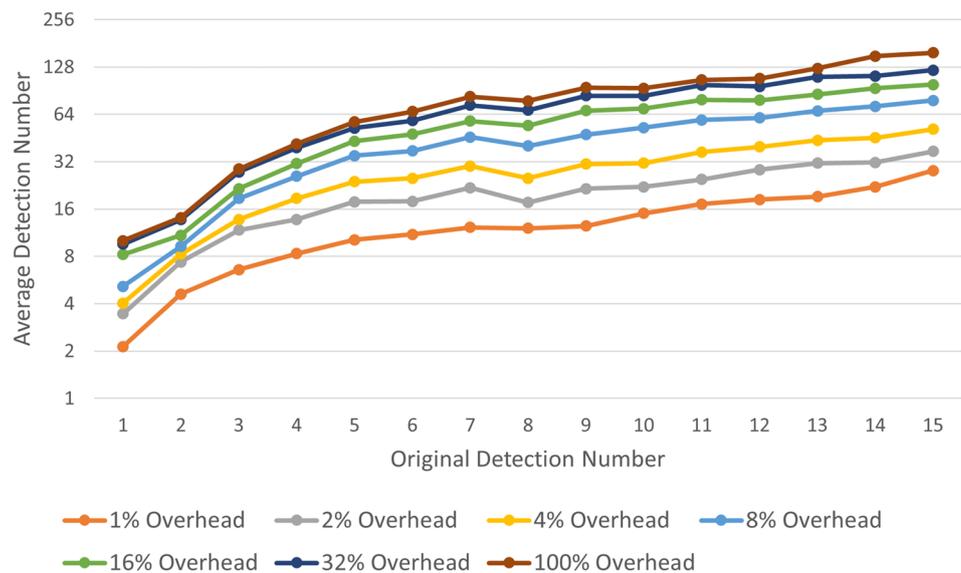


(c) Fm\_rec



(d) Colorconv

Fig. 4 (continued)



(e) Fpu

comparing the effect of the two types of patterns, a larger percent of low-detected faults gained extra detections when simulated with intermediate patterns than random patterns most of the time, and the results were very close in the other cases when the percentages were not larger, except in the case of Fpu.

This is reasonable when we consider that PI (Primary Input) settings of previous stuck-at ATPG patterns were kept the same when shifting a new pattern in scan chain, i.e. PI settings of intermediate patterns are the same as those of actual stuck-at ATPG patterns. However, the PI settings for random patterns were truly random. Even though the intermediate patterns are random-like due to the fact that each intermediate pattern is a combination of part of the captured result values that are being shifted out and part of the new pattern being shifted in, keeping the same ATPG PI settings make intermediate patterns more capable of detecting faults than random patterns. Thus, with random patterns, the testability of faults with the inclusion of the MISR structure can be shown before generating intermediate patterns for a new circuit, and a higher percent of improved low-detected faults can usually be expected with later intermediate pattern simulation.

#### 4.5 Summary Comparison of the Four Flip-Flop Selection Methods

In Fig. 3, all circuits with every flip-flop selection method and both types of simulated patterns can see faults obtain extra detections to various degrees. Among all flip-flop selection methods, Method 2 was able to help all circuits

obtain the largest percent of improved faults for all flip-flop overheads larger than 1%, and for some circuits (e.g. Quadratic, Des56, Colorconv) this was true even with 1% overhead. This may be due to:

1. Using an alternative model for fault detection analysis during flip-flop selection (e.g. cell-aware faults for Method 1) can lead to less effective results than using the targeted fault. (e.g. Here the targeted faults are stuck-at faults.)
2. The fault cone results in Methods 3 and 4 only consider circuit structure in flip-flop selection, but no pattern stimulus. There is no guarantee that a fault in a flip-flop's fault cone will be detected with an arbitrary pattern stimulus.

Our Fpu results demonstrate that not only the primary input values of the intermediate patterns, but other circuit or fault characteristics can matter when trying to improve the number of fault detections. In Fig. 3e, a larger difference can be seen for the percentage of improved faults with intermediate patterns compared to random patterns than for any other circuit. Moreover, for Fpu, even when 100% of the flip-flops were shadowed in the MISR, not all faults could gain extra detections. This is consistent with the results for cell-aware faults reported in [50, 51] for this circuit. Stuck-at ATPG patterns and intermediate patterns could not cover all cell-aware faults of Fpu fortuitously; thus, a non-negligible number of cell-aware ATPG top-off patterns needed to be added to guarantee full cell-aware fault coverage. One possible reason could be due to the character

of the fault detection requirements in Fpu. In particular, there may simply be more faults with difficult excitation or observation requirements that must be satisfied for fault detection to occur. This would make fortuitous detection with the “somewhat random” intermediate patterns much harder and would also make good assignments to the primary inputs potentially much more important.

As for Fm\_rec and Des56, their simulation results are two extremes. Fm\_rec can improve the detection number of most stuck-at faults with a small percent of flip-flops included in the MISR (after reaching 2% overhead in Fig. 3c), while Des56 requires a large overhead (Fig. 3b). This was also true for cell-aware fault detection in [50]. Only 2.2% flip-flops were required to be shadowed in the MISR for maximum cell-aware fault coverage in Fm\_rec, however, the highest percentage, 35.23% was required for Des56. Further fault cone analysis will be done in next section to explain this phenomenon.

#### 4.6 Fault Cone Analysis for Experimental Results and Circuit Characteristics

From the results obtained in [50, 51], we noticed that Des56 and Fm\_rec were two extremes with respect to the percentages of flip-flops that were required to be shadowed in the MISR to obtain maximum cell-aware fault coverage. Similar observations could be made for the data involving extra detections of stuck-at faults, as shown in Fig. 3b, c. In this section, we explore how fault cone analysis can reveal characteristics of the circuits regarding the relationship between flip-flops and fault detection. We will show that these circuit characteristics have a large impact on the amount of flip-flop overhead required to achieve high coverage with a MISR that captures test data during scan shift.

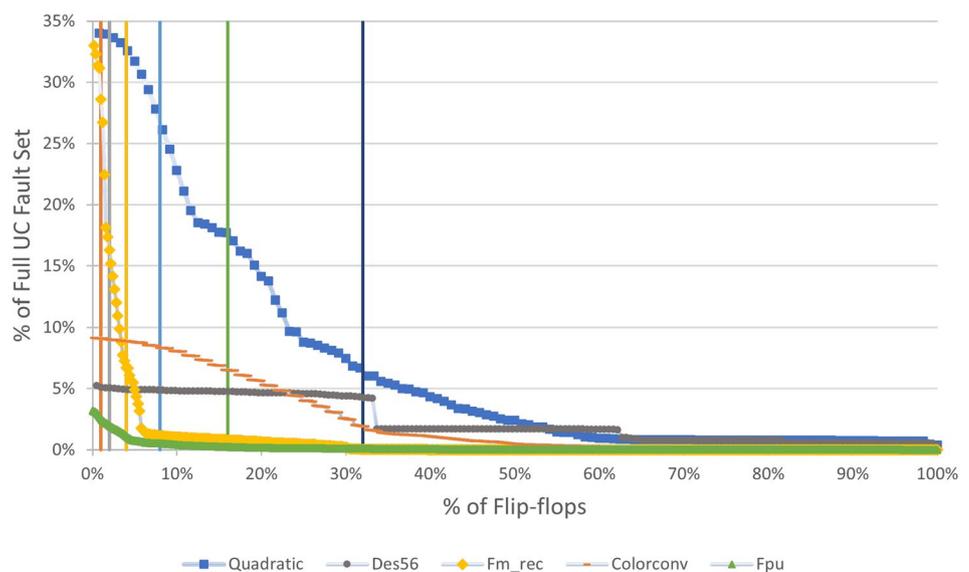
Figure 5 shows the fault cone analysis results. For each circuit, faults were reported to the D pin of each flip-flop by a commercial software tool. As described previously, among the six types of fault classes reported, only UC faults are likely to be detectable with pattern stimulus (although the detection is not guaranteed because fault detection still requires simultaneous excitation and observation by the patterns, and this is not guaranteed by the commercial tool when it reports fault cone data). Therefore, when analyzing fault cone results, only UC faults were counted. Intuitively, when more faults are identified as UC faults for a flip-flop, more faults have that flip-flop in their downstream cones of influence, and thus more fault detections are likely possible at that flip-flop. In contrast, a flip-flop with a small number of UC faults is likely to have few faults that can propagate to it.

In Fig. 5, the x-axis is the flip-flop percentage where the flip-flops are sorted in the order of the percentage of UC faults reported to the flip-flop from maximum to minimum. Dots were plotted based on the UC fault data for each flip-flop.

The values plotted on the y-axis show the percentage of all faults that are in the fault cone of each flip-flop. In other words, each dot corresponds to the number of faults reported as UC for a flip-flop divided by the number of faults in the full UC fault list. The larger the value plotted on the y-axis for a certain flip-flop, the more likely that flip-flop is to be capable of detecting more faults. Six vertical lines were added to the figure to show the sampled flip-flop percentages used in our previous experiments (1%, 2%, 4%, 8%, 16% and 32%).

In Fig. 5, we can see that, for Fm\_rec, only a small percentage of flip-flops contribute to most of the fault detection. Each flip-flop in the top 3.19% of the flip-flops has a

**Fig. 5** Per Flip-Flop UC Fault Coverage with 1%, 2%, 4%, 16% and 32% Flip-flop Vertical Lines



large number of faults in its fault cone. That could explain why Fm\_rec can get a large number of fault detections during scan shift with only a very small percentage of flip-flops shadowed in the MISR. In contrast, the highest fault cone percentages for Des56 are around 5%. The fault cones appear to be more evenly distributed among many flip-flops. This helps explain why Des56 requires a MISR with many shadow flops included to get good fault coverage across the circuit during scan shift.

Note that this means that Fm\_rec is more amenable to a low cost implementation of this approach than Des56. This is true even though Des56 is considerably smaller than Fm\_rec. This is not the first time that we have seen larger circuits behave better than smaller circuits for scan shift capture. We saw similar results when detecting cell-aware faults in [51], where circuit size was not a good indicator of the effectiveness of the approach. Fault cones help to determine some of the characteristics that actually are important.

## 5 Conclusion

In this paper, we ran simulation experiments with four flip-flop selection methods and two types of simulation patterns for extra detections of low-detected stuck-at faults. Our experimental results showed that selecting the flip-flops from a ranked list sorted based on the maximum stuck-at fault coverage obtained at each flop can highly improve the percentage of low-detected stuck-at faults that gain extra detections with the MISR structure. The detection count increases were higher than that achieved with the other three methods (including Method 1 introduced in [51]). The larger percentage of improved faults is significant. This analysis is important because larger the number of detections of the otherwise least detected stuck-at faults, the more likely a test is to fortuitously detect other untargeted faults or unmodeled defects. Also, a useful approximation of the expected benefit to be achieved by adding a MISR for scan shift detection appeared to be obtained by using random pattern simulation. Thus, random pattern simulation can be run as a quick test when the real intermediate patterns are not readily available yet. Note that our approach can be used in concert with other ATPG-based dynamic compaction approaches that aim to get high fault coverage, including high n-detect coverage. Our approach is also capable of being used in concert with an on-chip decompressor.

Furthermore, the influence of fault cone size on the experimental results was studied. This type of analysis can be very useful to a designer trying to decide whether to insert the proposed MISR structure into a

design. Specifically, if a few flip-flops have fault cones that together cover most of the faults, then the proposed approach may be a relatively low-cost way to improve coverage, reduce test time, and increase quality. On the other hand, if the faults are distributed relatively evenly into fault cones with limited overlap such that most of the flip-flops would need to be shadowed, a large MISR will likely be necessary to get high coverage.

**Funding** Open access funding provided by SCEL, Statewide California Electronic Library Consortium. This project was supported in part by National Science Foundation (NSF) grant CCF1812777 and CCF1814928. The authors have no additional competing interests to declare that are relevant to the content of this article.

**Data Availability** The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Alves N, Dworak J, Bahar I, Nepal K (2009) Compacting test vector sets via strategic use of implications. In: Proceedings of IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, pp 83–88
2. Blanton RD, Dwarakanath KN, Shah AB (2003) Analyzing the effectiveness of multiple-detect test sets. In: Proceedings of International Test Conference (ITC), pp 876–885
3. Dietrich C, Schmider A, Pusz O, Vayá GP, Lohmann D (2018) Cross-layer fault-space pruning for hardware-assisted fault injection. In: Proceedings of Design Automation Conference (DAC), San Francisco, CA, pp 1–6
4. Donglikar S, Banga M, Chandrasekar M, Hsiao MS (2009) Fast circuit topology based method to configure the scan chains in Illinois Scan architecture. In: Proceedings of International Test Conference (ITC), Austin, TX, USA, pp 1–10
5. Dworak J, Grimaila MR, Lee S, Wang L-C, Mercer MR (1999) Modeling the probability of defect excitation for a commercial IC with implications for stuck-at fault-based ATPG strategies. In: Proceedings of International Test Conference, Atlantic City, NJ, pp 1031–1037
6. Dworak J, Grimaila MR, Lee S, Wang L-C, Mercer MR (2000) Enhanced DO-RE-ME based defect level prediction using defect site aggregation - MPG-D. In: Proceedings of International Test Conference, Atlantic City, NJ, pp 930–939
7. Dworak J, Wicker JD, Lee S, Grimaila MR, Mercer MR, Butler KM, Stewart B, Wang L-C (2001) Defect-oriented testing and defective-part-level prediction. IEEE Des Test Comput 18(1):31–41

8. Dworak J, Wingfield J, Cobb B, Lee S, Wang LC, Mercer MR (2002) Fortuitous detection and its impact on test set sizes using stuck-at and transition faults. In: Proceedings of 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp 177–185
9. Dworak J, Wingfield J, Mercer MR (2004) A preliminary investigation of observation diversity for enhancing fortuitous detection of defects. In: Proceedings of 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), pp 460–468
10. Gao Z, Hu M, Swenton J, Malagi S, Huisken J, Goossens K, Marinissen EJ (2019) Optimization of cell-aware ATPG results by manipulating library cells' defect detection matrices. In: Proceedings of IEEE International Test Conference in Asia (ITC-Asia), Tokyo, Japan, pp 91–96
11. Gao Z, Malagi S, Hu M, Swenton J, Baert R, Huisken J, Chehab B, Goossen K, Marinissen EJ (2019) Application of cell-aware test on an advanced 3nm CMOS technology library. In: Proceedings of International Test Conference (ITC), pp 1–6
12. Gao Z, Malagi S, Marinissen EJ, Swenton J, Huisken J, Goossens K (2019) Defect-location identification for cell-aware test. In: Proceedings of IEEE Latin American Test Symposium (LATS), Santiago, Chile, pp 1–6
13. Geuzebroek J, Marinissen EJ, Majhi A, Glowatz A, Hapke F (2007) Embedded multi-detect ATPG and its effect on the detection of unmodeled defects. In: Proceedings of International Test Conference (ITC), Santa Clara, CA, pp 1–10
14. Grimaila MR, Lee S, Dworak J, Butler KM, Stewart B, Balachandran H, Houchins B, Mathur V, Park J, Wang L, Mercer MR (1999) REDO-random excitation and deterministic observation-first commercial experiment. In: Proceedings of 17th IEEE VLSI Test Symposium, Dana Point, CA, pp 268–274
15. Gupta S, Crouch A, Dworak J, Engels D (2017) Increasing IJTAG bandwidth and managing security through parallel locking-SIBs. In: Proceedings of IEEE International Test Conference (ITC), Fort Worth, TX, USA, pp 1–10
16. Hapke F, Krenz-Baath R, Glowatz A, Schloeffel J, Hashempour H, Eichenberger S, Hora C, Adolfsson D (2009) Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs. In: Proceedings of International Test Conference (ITC), pp 1–10
17. Hapke F, Redemund W, Glowatz A, Rajsaki J, Reese M, Hustava M, Keim M, Schloeffel J, Fast A (2014) Cell-aware test. *IEEE Trans Comput Aided Des Integr Circ Syst* 33(9):1396–1409
18. Hapke F, Reese M, Rivers J, Over A, Ravikumar V, Redemund W, Glowatz A, Schloeffel J, Rajsaki J (2021) Cell-aware Production test results from a 32-nm notebook processor. In: Proceedings of International Test Conference (ITC), pp 1–9
19. Hapke F, Schloeffel J (2012) Introduction to the defect-oriented cell-aware test methodology for significant reduction of DPPM rates. In: Proceedings of IEEE European Test Symposium (ETS), pp 1–6, 28
20. Home: OpenCores. [Online]. Available: <http://opencores.org/>. Accessed 30 Aug 2016
21. Hu M, Gao Z, Malagi S, Swenton J, Huisken J, Goossens K, Wu C, Marinissen EJ (2020) Tightening the mesh size of the cell-aware ATPG net for catching all detectable weakest faults. In: Proceedings of IEEE European Test Symposium (ETS), Tallinn, Estonia, pp 1–6
22. Huang Y (2006) On N-detect pattern set optimization. In: Proceedings of 7th International Symposium on Quality Electronic Design (ISQED'06), San Jose, CA, pp 6–450
23. Hunger M, Hellebrand S, Czutro A, Polian I, Becker B (2009) ATPG-based grading of strong fault-secureness. In: Proceedings of 15th IEEE International On-Line Testing Symposium, Sesimbra, Lisbon, pp 269–274
24. Jiang H, Zhang F, Sun Y, Dworak J (2018) One more time! Increasing fault detection with scan shift capture. In: Proceedings of IEEE 27th North Atlantic Test Workshop (NATW), Essex, VT, pp 1–7
25. Kantipudi K (2007) Minimizing N-detect tests for combinational circuits. Ph.D. Dissertation, Auburn University
26. Kantipudi KR, Agrawal VD (2006) On the size and generation of minimal N-detection tests. In: Proceedings of 19th International Conference on VLSI Design (VLSID), pp 425–430
27. Lee S, Cobb B, Dworak J, Grimaila MR, Mercer MR (2002) A new ATPG algorithm to limit test set size and achieve multiple detections of all faults. In: Proceedings of Design, Automation and Test in Europe Conference (DATE), Paris, pp 94–99
28. Lin YT, Ezekwe CU, Blanton RD (2009) Physically-aware n-detect test relaxation. In: Proceedings of VLSI Test Symposium, pp 197–202
29. Liu C, Chakrabarty K (2003) A partition-based approach for identifying failing scan cells in scan-BIST with applications to system-on-chip fault diagnosis. In: Proceedings of Design, Automation and Test in Europe Conference (DATE), Munich, Germany, pp 230–235
30. Liu C, Chakrabarty K (2004) Identification of error-capturing scan cells in scan-BIST with applications to system-on-chip. *IEEE Trans Comput Aided Des Integr Circ Syst* 23(10):1447–1459
31. Lorenzelli F, Gao Z, Swenton J, Malagi S, Marinissen EJ (2021) Speeding up cell-aware library characterization by preceding simulation with structural analysis. In: Proceedings of IEEE European Test Symposium (ETS), Bruges, Belgium, pp 1–6
32. Ma S, France P, McCluskey EJ (1995) An experimental chip to evaluate test techniques: Experimental results. In: Proceedings International Test Conference (ITC), pp 663–672
33. McCluskey EJ, Tseng C-W (2000) Stuck-fault tests vs. actual defects. In: Proceedings of International Test Conference (ITC), pp 336–342
34. Mukherjee N, Tille D, Sapati M, Liu Y, Mayer J, Milewski S, Moghaddam E, Rajsaki J, Solecki J, Tyszer J (2019) Test time and area optimized BIST scheme for automotive ICs. In: Proceedings of IEEE International Test Conference (ITC), Washington, DC, USA, pp 1–10. <https://doi.org/10.1109/ITC44170.2019.9000133>
35. Mrugalski G, Rajsaki J, Solecki J, Tyszer J, Wang C (2015) TestExpress - new time-effective scan-based deterministic test paradigm. In: Proceedings of IEEE Asian Test Symposium (ATS), pp 19–24
36. Neophytou S, Michael MK (2008) On the relaxation of n-detect test sets. In: Proceedings of 26th IEEE VLSI Test Symposium (VTS), San Diego, CA, pp 187–192
37. Neophytou S, Michael MK, Christou K (2009) Generating diverse test sets for multiple fault detections based on fault cone partitioning. In: Proceedings of 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Chicago, IL, pp 401–409
38. Pomeranz I (2012) Non-uniform coverage by n-detection test sets. *IEEE Trans Very Large Scale Integr* 20(11):2138–2142
39. Pomeranz I, Reddy SM (1999) On n-detection test sets and variable n-detection test sets for transition faults. In: Proceedings of 17th IEEE VLSI Test Symposium (VTS), pp 173–180
40. Pomeranz I, Reddy SM (2001) On the use of fault dominance in n-detection test generation. In: Proceedings of 19th IEEE VLSI Test Symposium (VTS), pp 352–357
41. Pomeranz I, Reddy SM (2004) A measure of quality for n-detection test sets. *IEEE Trans Comput* 53(11):1497–1503
42. Pomeranz I, Reddy SM (2008) On the Saturation of n-detection test generation by different definitions with increased n. In: *IEEE Transactions Computer-Aided Design Integr. Circ Syst* 27(5):946–957
43. Rajsaki J, Tyszer J, Kassab M, Mukherjee N (2004) Embedded deterministic test. *IEEE Trans Comput Aided Des Integr Circ Syst* 23(5):776–792

44. Sultana S, Fekr AR, Radecka K (2013) SAT-based reversible gate/wire replacement fault testing. In: Proceedings of 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), Columbus, OH, pp 1075–1078
45. Sun Y, Zhang F, Jiang H (2019) Repurposing FPGAs for tester design to enhance field-testing in a 3D stack. *J Electron Test* 35:887–900
46. Tsai K, Gopalakrishnan S (2017) Test coverage analysis for designs with timing exceptions. In: Proceedings of IEEE 26th Asian Test Symposium (ATS), pp 169–174
47. Wang Z, Marek-Sadowska M, Tsai KH, Rajski J (2003) Multiple fault diagnosis using n-detection tests. In: Proceedings of 21st International Conference on Computer Design, pp 198–201
48. Wang Z, Marek-Sadowska M, Tsai KH, Rajski J (2006) Analysis and methodology for multiple-fault diagnosis. *IEEE Trans Comput Aided Des Integr Circ Syst* 25(3):558–575
49. Yu C, Reddy SM, Pomeranz I (2004) Weighted pseudo-random BIST for n-detection of single stuck-at faults. In: Proceedings of Asian Test Symposium, pp 178–183
50. Zhang F, Dworak J (2018) Cell-aware fault analysis and test set optimization in digital integrated circuits
51. Zhang F, Hwong D, Sun Y, Garcia A, Alhelaly S, Shofner G, Winemberg L, Dworak J (2016) Putting wasted clock cycles to use: Enhancing fortuitous cell-aware fault detection with scan shift capture. In: Proceeding of International Test Conference (ITC), pp 1–9
52. Zhang F, Thornton M, Dworak J (2014) When optimized N-detect test sets are biased: an investigation of cell-aware-type faults and N-detect stuck-at ATPG. In: Proceedings of IEEE 23rd North Atlantic Test Workshop, Johnson City, NY, pp 32–39

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Hui Jiang** received her Bachelor of Science degree in Electronic Engineering from Shandong University in 2014, Master of Science degree in Computer Engineering from Southern Methodist University in 2016, and Doctor of Philosophy degree in Computer Engineering from Southern Methodist University in 2022. Her research focuses on optimizing test sets and Design for Testability (DFT) circuitry to achieve high test quality with low test cost. In particular, her research

involves exploring how test clock cycles can be most efficiently utilized to provide efficient detection of stuck-at faults and transition faults. She interned at Qualcomm Inc. in Product Test Engineering team during Summer of 2020 and Summer of 2021. Since Jan. 2022, Hui has joined Cadence Design Systems, Inc. as a Lead Software Engineer, DFT.

**Fanchen Zhang** received his B.S. degree in Telecommunication Engineering from Harbin Institute of Technology in Harbin, Heilongjiang province, China, M.S. degree in electrical engineering, and PhD degree in computer engineering from Southern Methodist University in Dallas, TX, USA in 2001, 2012 and 2018, respectively. He is currently a Hardware Engineer at Cisco Systems, Inc. in San Jose, CA, USA. His work includes semiconductor Design-for-Test (DFT) architecture, scan insertion, Memory Built-In Self Test (MBIST) implementation, pre-silicon verification tests, post-silicon validation and debug.

**Jennifer Dworak** received her B.S., M.S., and Ph.D. degrees in electrical engineering from Texas A&M University in College Station, TX. She is currently a Professor of Electrical & Computer Engineering at Southern Methodist University and is the Associate Director of the SMU AT&T Center for Virtualization. Her research interests include manufacturing test, reliability, and hardware security of integrated circuits and digital systems.

**Kundan Nepal** received the B.S. degree in electrical engineering from Trinity College, Hartford, CT, USA, the M.S.E.E. degree from the University of Southern California, Los Angeles, CA, USA, and the Ph.D. degree in electrical and computer engineering from Brown University, Providence, RI, USA, in 2002, 2003, and 2007, respectively. He is currently a Professor of Electrical & Computer Engineering at the University of St. Thomas, St Paul, MN, USA. His current research interests include embedded systems, defect/fault tolerant circuits and systems, nanometer digital VLSI system design, and reconfigurable computing.

**Theodore Manikas** received the B.S. degree in electrical engineering from Michigan State University, the M.S. degree in electrical engineering from Washington University (St. Louis) and the Ph.D. degree in electrical engineering from the University of Pittsburgh. He is currently a Clinical Professor in the Department of Computer Science at Southern Methodist University, Dallas, TX, USA. His current research interests include computer systems design, security, and testing. He is a Licensed Professional Engineer in Texas and Oklahoma.