LS-TDF: Low-Switching Transition Delay Fault Pattern Generation*

Jeremy Lee and Mohammad Tehranipoor
ECE Department
University of Connecticut
{jslee,tehrani}@engr.uconn.edu

***Abstract*— Higher chip densities and the push for higher performance have continued to drive design needs. Transition delay fault testing has become the preferred method for ensuring these chips operate at the specified frequency. However, current industrially used X-filling (random-fill or compression) schemes tend to generate transition delay fault patterns with switching activity much higher than what would be seen during functional mode operation of the chip, potentially causing failures that would not occur in the field. In this paper, we present a low-switching transition delay fault pattern generation flow. The flow short-lists patterns based on high switching activity, which is determined by the fault lists of each pattern. Once those patterns with high switching are filtered, they will be replaced by low-switching patterns to recover any lost fault coverage. The proposed pattern generation flow works well with commercial tools and can easily be integrated into an industrial flow.**

I. Introduction

Supply noise effects have become much more significant in the design and test of nanometer technology designs within recent years. This is due to the fact that the threshold voltage is not decreasing proportionally as supply voltage scales, resulting in reduced noise immunity and increased circuit sensitivity. Excessive power supply noise leads to increased delays, hindering performance, or functional failures.

Testing nanometer scale, high-performance designs at functional clock speeds to ensure correct field operation continue to be an increasingly large challenge, especially due to the increasing presence of these effects. Transition delay fault (TDF) testing has been widely used to ensure fabricated chips meet the designed timing specifications [1] [2] [3] [4] and to detect any timing-related failures. However, in order to maintain adequate transition fault test coverage in a reasonable amount of time, ATPG tools have traditionally tried to maintain algorithms with low overhead. This presents drawbacks since these algorithms often are not timing- and power-aware and do not consider delays due to crosstalk or supply noise.

Currently, ATPGs deterministically target a transition fault then either perform random filling to increase fortuitous detection of unmodeled faults or use compression algorithms to reduce the total pattern count. In either case, switching activity is known to significantly increase beyond the expected average functional switching [5] [6] [7] [8] [9]. Since the chip was not designed to expect such power demands, these TDF patterns generated can potentially fail the chip during test due to excessive IR-drop but still operate correctly in the field, i.e. causing yield loss. This has driven the need for new low-switching TDF pattern generation methods that can reliably detect defects in bad chips while not over-testing good chips.

A. Related Prior Works

A pattern post-processing technique (i.e. pattern validation) to verify TDF patterns that cause excessive IR-drop is proposed in [6]. Attempting to address this verification in dynamic simulation will force the use of circuit simulation or mixed-level simulation techniques, which are expensive in terms of run time. Also, a method of measuring average power called switching cycle average power (SCAP) was proposed in [8] to produce supply noise tolerant TDF patterns. SCAP considers both simultaneous switching and affected long paths. The method requires pattern delay information which may make it computationally intensive.

Additionally, a vector-based compaction solution to reduce overkill and power supply noise induced delay has been proposed in and [10]. The authors developed a vector-dependent power supply noise analysis solution that models the voltage drop based on the layout of the chip. The preferred-fill technique proposed in [7], attempts to reduce the Hamming distance between the initialized, launched, and captured patterns during TDF testing. In [9], the authors propose a low-capture-power (LCP) X-filling method for assigning 0s and 1s to the X-bits in a test cube so that the number of transitions at the outputs of scan flip-flops in capture mode for the resulting fully-specified test vector is reduced. The authors in [11] propose another method, called capture-aware (CA) test cube generation, for deterministically generating test cubes not only for fault detection but also for capture power reduction. Most other previously propose methods focus on power reduction during shift operation.

B. Contribution and Paper Organization

In this paper, we propose a novel use for the TDF pattern fault lists. By observing the number of faults detected and using a statistical ratio of those faults still left unobservable by a pattern, a measure comparable to the pattern *weighted switching activity* (WSA) can be used to validate TDF patterns that maintain switching near or below average functional switching activity and short-list those that exceed the given threshold. The initial pattern set can be generated with any ATPG tool and perform X-filling to achieve the desired compression or coverage, then validated with this technique. Once the new short-listed pattern set is validated, the lost coverage must then be compensated with a second phase pattern generation with an alternative low-switching pattern generation technique.

The remainder of this paper is organized as followed. Section II explains how the switching activity is inferred from

*This work was supported in part by Semiconductor Research Corporation under contracts No. 1455 and 1587.

1093-0167/08 $25.00 © 2008 IEEE
DOI 10.1109/VTS.2008.48

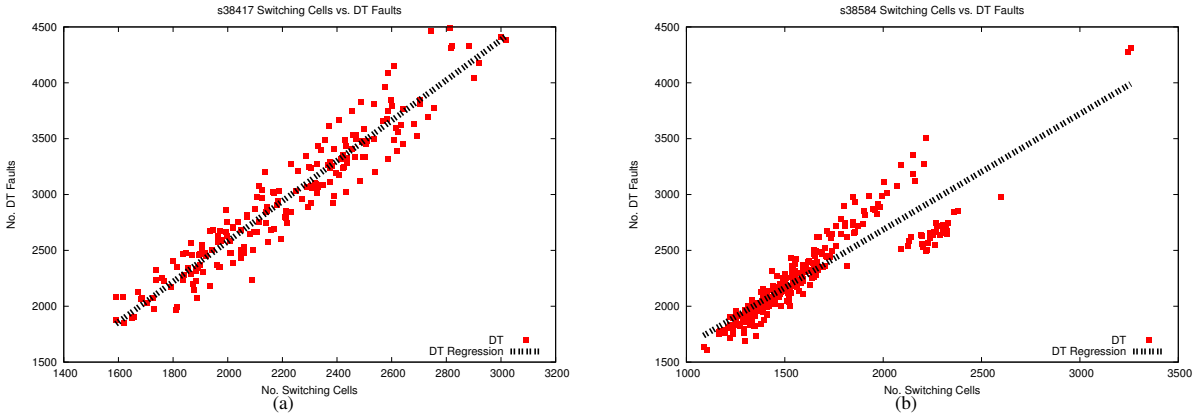

Fig. 1. (a) The number of DT faults of the pattern set for s38417 is closely correlated to the gate switching activity. (b) The switching activity for s38584 cannot be inferred simply by the DT fault list due to switching occurring at NO fault sites, especially as the number of detected faults increase.

the fault lists of each TDF pattern. In Section III, the pattern generation flow for low-switching transition delay faults (LS-TDF) is presented. Section IV discusses the results of the LS-TDF flow on several ISCAS89 benchmarks. Finally, we conclude our discussion in Section V.

II. Inferring Switching Activity

Previous solutions to reduce switching activity during shift operation or launch-to-capture cycle have often relied on calculating the number of gates switching, weighted switching activity (WSA), or simply transitions occurring at the flip-flops [8] [9].We propose using the fault list as a measure to calculate the switching activity created by the pattern set during the launch-to-capture cycle, which will later be used to classify patterns with excessive switching activity.

By using the same fault list used by the ATPG, it is possible to correlate the switching activity created by the pattern. During transition delay fault ATPG, the ATPG engine is targeting fault sites where it is possible to detect a rising or falling transition. So, when it is detected, a generated pattern will cause a transition starting from a controllable input (also called *launch-point*) and propagate to an observable output (also called *end-point*). Launch-points are either scan-flops or primary inputs where a transition is initially started, while end-points (either scan-flops or primary output) are where the result of the transition is finally captured. In addition to gate switching at the detected fault site, all other gates along the path between the launch-point and end-point also experience a transition, which will also be considered detected faults.

Faults reported by the ATPG engine as detected (DT) are those that it was able to stimulate the fault site with both 0 and 1 and generate and propagate a transition from a launch-point through the fault site and to an end-point. In addition to DT faults, there are also faults that are not detectable through fault simulation with the current pattern. These not detectable (ND) faults fall into two potential categories: not controllable (NC) and not observable (NO). NC transition faults are those faults the ATPG is not able to control either to a 0 or a 1 state at their respective fault site. While NO transition faults are those faults where both states are controllable at that fault site but either a transition was not generated at the site or it was not possible to propagate the generated transition to an observable end-point [12] [13]. For instance, in benchmark s38417, the ATPG [12] reports the following results for a single pattern of the pattern set: DT=3748, ND=36686, NO=27148, NC=9538, where ND = NC + NO.

Each pattern, due to deterministic pattern generation and X-filling, is likely to detect many transition faults at a time that are not on the targeted path between the launch-point and end-point, creating significant observable switching throughout the chip. In some cases, the switching activity is closely correlated to the number of detected (DT) faults in a circuit as shown for ISCAS'89 benchmark s38417 in Figure 1(a). However, all switching is not limited to just those faults that are detected, which can be seen in Figure 1(b) for benchmark s38584. For s38584, the number of DT faults begin to diverge as the number of cells switching increase, preventing DT from being a reliable measure of cells switching. This is due to switching activity at fault sites that are reported as NO by the tool.

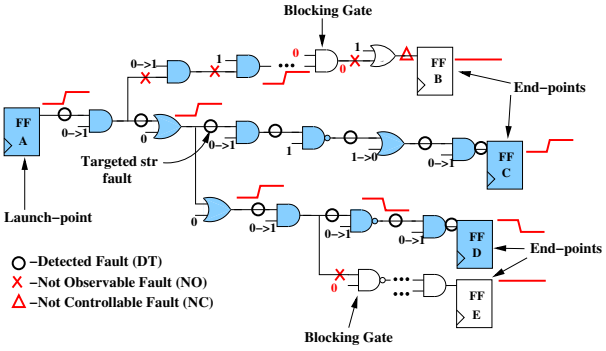

Fig. 2. The pattern generated detects faults along the *AC* and *AD*, which can represent switching activity. There is still switching activity on other paths, but transitions do not reach an end-point and are included in the NO fault list.

To infer the switching activity from the transition delay fault list, the NO fault list must also be considered since a transition likely occurred, but due to the manner some don't-care states were filled, the transition is blocked in a fan-out path. This is demonstrated in Figure 2. The *str* fault is launched from launch-point *A*, stimulating transitions through paths *AC* and *AD* to be observed at end-points *C* and *D*, while adding all of those faults to the detected fault list, there are still transitions

being propagated through fan-out paths *AB* and *AE* for some distance until both are obstructed by off-path inputs. These off-path inputs may be in this state to correctly propagate the transition through the *AC* or *AD* or as a result of X-filling. So, while some of the NO faults create transitions on these off-paths, other NO faults do not transition at all, like those faults that lie beyond the blocking gates.

In order to account for both of these factors and create an accurate description of the switching activity, we introduce Fault List Inferred Switching (FLIS). FLIS correlates the DT faults and NO faults to the switching activity created by the associated pattern. Since not all NOs can be considered switching, a statistical measure of the likelihood of a transition occurring at that fault site is used.

The transition propagation probability (TPP) measure is proposed to determine the probability of a transition occurring at these NO fault sites. It is based on the topology and logic of the circuit. To reduce complexity and keep probability calculation to a single pre-processing step, the TPP is first calculated independent of the applied input pattern. So, although the FLIS is dependent on the fault lists of each pattern, the NO fault weight used to calculate the FLIS is based on pre-calculated TPP values.

Figure 3(a) shows an example of the TPP calculation. The TPP for each fault site is a 4-tuple (P_{S0}, P_{S1}, P_R, P_F) specifying the probabilities of a steady-state 0 (P_{S0}), steady-state 1 (P_{S1}), rising transition (P_R), and falling transition (P_F) at the output of the gate. Since this is a pattern independent calculation, all primary inputs and scan-flops start with the 4-tuple (0.25, 0.25, 0.25, 0.25). The equations to propagate these probabilities depend on the logic function and number of inputs. For example, a 2-input AND gate (A and B are the inputs and C is the output) would calculate P_{CS0} (probability of steady-state 0 at output C) as shown in Equation 1.

$$\begin{aligned} P_{CS0} \quad = \quad & P_{AS0}P_{BS0} + P_{AS0}P_{BS1} + P_{AS1}P_{BS0} + \\ & P_{AS0}P_{BR} + P_{AS0}P_{BF} + P_{AR}P_{BS0} + \\ & P_{AF}P_{BS0} + P_{AR}P_{BF} + P_{AF}P_{BR} \end{aligned} \tag{1}$$

Probabilities P_{CS1}, P_{CR}, and P_{CF} are solved in a similar manner. The TPP for each gate then propagates to the next gate in the fan-out cone and used as the input probabilities for the next gate. Once the probabilities for all gates are calculated, determining the probability of a transition (P_{TR}) occurring at a fault site is equal to $P_R + P_F$.

FLIS calculation is dependent on the reported DT and NO faults by ATPG. While DT faults fully contribute to the FLIS calculation, NO faults only contribute a percentage of their weight to the FLIS according to the TPP (i.e. P_{TR}). The total FLIS$_i$ for each individual pattern can be calculated as shown in Equation 2.

$$FLIS_i = N_{DTi} + \sum_{j=0}^{N_{NOi}} P_{TRj}, \tag{2}$$

where N_{DTi} is the number of detected faults by pattern i, N_{NOi} is the number of NO faults reported for that pattern, and P_{TRj} is the probability of a transition (based on the TPP) occurring at that NO fault site.

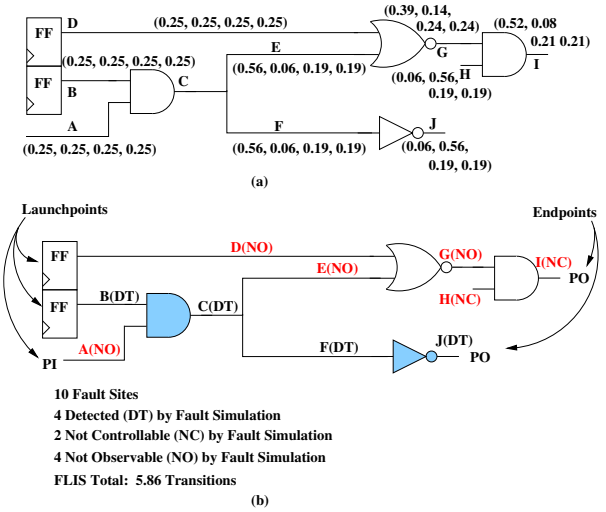

Fig. 3. (a) Pre-processing the transition propagation probability calculation. (b) Faults detected (DT) count as full transitions in FLIS calculation while NO faults only contribute a portion of their total transitioning potential to FLIS.

After fault simulating an arbitrary transition delay pattern for the circuit in Figure 3(a), the fault report will specify those that are DT, NC, and NO, as shown in Figure 3(b). In this figure, all cases (DT, NO, NC) are reported by the ATPG and are used to calculate FLIS. Since NC faults are not controllable to either a 0 or 1-state, the possibility of a transition at those fault sites are excluded. The FLIS calculation for the example circuit shown below will occur as follows:

1) Nets B, C, F, J: $P_{BTR} = P_{CTR} = P_{FTR} = P_{JTR} = 1$
2) Net A: $P_{ATR} = P_{AR} + P_{AF} = 0.5$
3) Net D: $P_{DTR} = P_{DR} + P_{DF} = 0.5$
4) Net E: $P_{ETR} = P_{ER} + P_{EF} = 0.19 + 0.19 = 0.38$
5) Net G: $P_{GTR} = P_{GR} + P_{GF} = 0.24 + 0.24 = 0.48$
6) Net H, I: $P_{HTR} = P_{ITR} = 0$

- Total FLIS: 5.86

The complexity of the FLIS calculation depends on the number of gates, number of NO faults, and number of patterns. These factors constitute a complexity of $O(N_{PT} \cdot N_{NO} \cdot N_g)$. For large circuits, $N_{PT} << N_{NO}$ and N_g, and is considered negligible compared to the other two variables. Also, $N_{NO} < N_g$, therefore the complexity will be $O(N_g^2)$.

With the FLIS calculation complete for this pattern, we can now determine whether or not the pattern creates high switching activity in the circuit. The pattern will be retained if it provides FLIS below the average gate switching activity. This will be done by correlating the FLIS to the gate switching activity. The results shown in Section IV will show this strong correlation between these two.

III. LS-TDF Pattern Generation Flow

LS-TDF pattern generation integrates the FLIS calculation introduced in Section II with existing commercial ATPG tools to prevent patterns from excessively exceeding average functional switching activity. This will prevent patterns from overexercising the chip beyond functional stress.

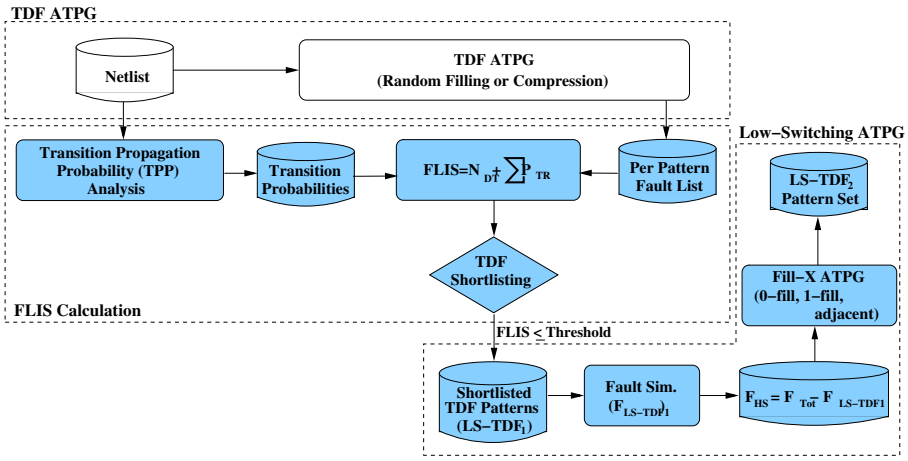

Fig. 4. The LS-TDF flow. The flow validates patterns that have low-switching activity and short-lists those that have high switching activity. LS-TDF = LS-TDF$_1$ + LS-TDF$_2$

The LS-TDF pattern generation flow is shown in Figure 4. LS-TDF can be divided into three parts: (1) TDF ATPG, (2) FLIS Calculation and Short-listing, and (3) Low-Switching ATPG.

A. TDF ATPG

The first phase of LS-TDF involves conventional TDF ATPG with any commercial tool. This will allow test engineers to use the tools to perform X-filling in their preferred method (random filling or compression) that complies with their original industrial test automation flows. Once the pattern generation is complete, each pattern is then passed through the fault simulator to determine DT, NC, and NO faults. The DT and NO fault lists will then be used as an input to the FLIS Calculation and Short-listing phase of the LS-TDF flow.

B. FLIS Calculation and Short-listing

The second phase requires both the transition propagation probabilities (TPP) and DT and NO fault lists. To calculate the TPP, again the gate level netlist will be necessary. The TPP can be generated in parallel with TDF ATPG and only has to be calculated once for a netlist since the probabilities are independent of the TDF test patterns.

Once the TPP is generated for all fault sites and the fault lists (DT and NO) from fault simulation is obtained, the FLIS can be calculated to obtain the switching activity created by each test pattern. This phase will validate the test patterns to ensure the switching activity does not exceed average functional switching activity.

Since the FLIS can be used as a measure similar to WSA, we can set a threshold at a comparable level to filter patterns with a FLIS greater than the threshold. This will allow validation of pattern switching activity without full logic simulation to determine the WSA. In other words, the low-switching TDF pattern generation will be significantly faster than WSA-based methods.

C. Low-Switching TDF ATPG

The final phase of the LS-TDF flow is to regenerate patterns for the coverage loss by short-listing the pattern set. In order to determine how much coverage was actually lost, the LS-TDF$_1$ patterns set, generated by short-listing, must be passed through the fault simulator once again to determine which faults are detected with this pattern set. Often during pattern generation, many patterns will detect many of the same faults. Although the patterns that were short-listed had a lot of switching activity, many of the faults may not necessarily be unique to them. By performing fault simulation on the LS-TDF$_1$ pattern set, we can determine how many faults have been omitted by the short-listing as summarized in Equation 3.

$$F_{HS} = F_{Tot} - F_{LS-TDF1} \quad (3)$$

F_{HS} denotes the faults excluded by the short-listed pattern set. These remaining faults are determined by the difference of the total fault set, F_{Tot}, and the faults detected by fault simulation of the LS-TDF$_1$ pattern set, $F_{LS-TDF1}$. This will allow the active fault list to be limited to F_{HS} during the second pass of the ATPG.

Since the conventional TDF ATPG flow used in the first phase created those patterns with high switching activity to detect those faults, a different manner of X-filling must be used for this second pass. Here, we perform 0-fill during TDF ATPG to reduce switching activity during test, however, performing alternative low-switching fill methods such as 1-fill and adjacent fill could be just as viable. The methods proposed in [7] [14] [15] could also be used in this flow but may conflict with existing compression methods. In the second pass, since the ATPG is working on a much shorter fault list, the pattern volume increase should not be as significant as if the entire fault list were active.

IV. Experimental Results

The LS-TDF flow presented in Section III was implemented on x86-based desktop PC with a 3-GHz processor and 1 GB of RAM running a Linux operating system. The flow was implemented on several ISCAS'89 benchmarks [16].

The TDF patterns were generated using Synopsys TetraMax [12] and the TPP measures and FLIS calculation were implemented using TCL scripts, which is natively supported by TetraMax. In order to validate the FLIS calculation, a Verilog PLI [17] written in C was used to interface with

TABLE I

COMPARISON OF ISCAS'89 BENCHMARK CORRELATION COEFFICIENTS BETWEEN DT FAULTS AND SWITCHING CELLS AND THE FLIS AND SWITCHING CELLS.

	Correlation Coefficient	
Benchmark	DT and SW Cells	FLIS and SW Cells
s1196	0.84	0.96
s1238	0.83	0.92
s5378	0.87	0.93
s9234	0.86	0.89
s13207	0.93	0.92
s15850	0.81	0.88
s35932	0.57	0.91
s38417	0.95	0.94
s38584	0.90	0.99

Synopsys VCS [12] to monitor the switching activity during simulation. With the switching activity collected from the PLI, the DT faults reported by the fault simulator, and the FLIS calculations, a correlation coefficient was calculated to relate both the number of DT faults and FLIS to the total switching activity in the circuit. Figures 5(a) and 5(b) plot the FLIS against the switching activity for benchmarks s38417 and s38584, which were in Section II to show that the number of DT faults cannot always be accurately used as a measurement for switching activity. Both figures demonstrate the close correlation between the FLIS and number of cells switching, especially when comparing Figure 5(b) to Figure 1(b).

Additional correlation analysis for the FLIS calculation and number of switching cells is shown in Table I. The ISCAS'89 benchmark is listed in the first column, while the correlation of cells switching with the number of DT faults and FLIS are listed in columns 2 and 3, respectively. For most of the benchmarks, the FLIS correlation coefficient is greater than the correlation between DT and the switching cells. The two exceptions are in benchmarks s13207 and s38417, however, the difference in coefficients is almost negligible. From these results, we can conclude that the FLIS can be used as an accurate measurement of the switching activity in a circuit induced by the pattern set.

Figures 6(a) and 6(b) show the results of short-listing the patterns based on the FLIS threshold. The threshold is set based on the average cells switching defined by the designer during power/ground network synthesis. Given the high correlation between the cells switching and FLIS, we can set the FLIS threshold at a value comparable to the *average functional switching*. In practice, designers consider 20-25% average functional switching in today's large designs which it will be about 30% FLIS based on our calculations. FLIS thresholds of 27%, 30%, and 33% have been used in our simulations. As seen in the figures, the 27% threshold eliminates a majority of the patterns from the pattern set. At 30%, in Figure 6(a), most of the patterns are still short-listed while very few are excluded in Figure 6(b). Finally, at 33% FLIS threshold, only those patterns with very high switching activity are excluded. More results are shown in Table II for additional benchmarks for the same three thresholds. For s35932, due to a small pattern set, all patterns generate very high switching activity and were excluded for all three thresholds. For s1196 and s1238 all patterns are retained due to the small size of the benchmarks and test power does not seem to be an issue.

TABLE II

SHORTLISTING BASED ON PERCENTAGE OF MAXIMUM FLIS FOR ISCAS'89 BENCHMARKS.

Benchmark	Total Patterns	LS-TDF$_1$		
		27%	30%	33%
s1196	31	31	31	31
s1238	27	27	27	27
s5378	109	93	109	109
s9234	158	135	157	158
s13207	135	114	132	135
s15850	108	87	107	108
s35932	28	0	0	0
s38417	202	0	69	183
s38584	269	202	264	267

Table III shows the results of the LS-TDF flow for a 30% FLIS threshold. The number of cells, including gates and flip-flops in the benchmarks are listed in column 2. Columns 3 and 4 have the number of patterns generated by TDF ATPG and the respective fault coverage. The fault coverage is from LOC TDF with PIs held constant and restricting the ATPG from PO measures (*no PI Change and no PO measure*). Column 5 shows the test pattern set average cell switching activity monitored by the PLI routine. Columns 6, 7, and 8 show the pattern counts for LS-TDF$_1$, LS-TDF$_2$, and total LS-TDF that are generated by the LS-TDF flow. The pattern increase due to the LS-TDF flow does not significantly effect pattern volume except in the case of s38417. This is likely due 0-fill not being able to effectively target faults. From our experiments, the LS-TDF flow runs relatively fast on the benchmarks.

V. CONCLUSION

In this paper, we have proposed a novel method for low-switching transition-delay fault test pattern generation called LS-TDF ATPG. Unlike previously proposed methods that rely on the number of switching or weighted switching activity (WSA), our proposed LS-TDF ATPG takes the number of detected (DTs) and not-observable (NO) faults into account for power analysis. The FLIS value obtained for various benchmarks using DT and NO has shown a strong correlation with number of cell switching. The method identifies the high-switching patterns and replaces them with low-switching ones. The results show slight increase in test pattern count while it heavily depends on the FLIS threshold set by designer. The proposed method is practice oriented and can be easily adopted in current industrial test flow and commercial ATPG tools.

REFERENCES

[1] J. Savir, "Skewed-Load Transition Test: Part I, Calculus," in *Proc. of Intl. Test Conference*, 1992, pp. 705–713.

[2] J. Savir and S. Patil, "On Broad-Side Delay Test," in *Proc. of VLSI Test Symposium*, 1994, pp. 284–290.

[3] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. Kumar, S. Basu, D. J. Campbell, and J. Berech, "Scan-Based Transition Fault Testing – Implementation and Low Cost Test Challenges," in *Proc. of Intl. Test Conf.*, Oct. 2002, pp. 1120–1129.

[4] X. Lin, R. Press, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli, "High-Frequency, At-Speed Scan Testing," *IEEE Design & Test of Computers*, vol. 20, no. 5, pp. 17–25, 2003.

[5] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, "A Case Study of IR-Drop in Structured At-Speed Testing," in *Proc. of Intl. Test Conf.*, Sept. 2003, pp. 1098–1104.

[6] A. Kokrady and C. P. Ravikumar, "Fast, Layout-Aware Validation of Test-Vectors for Nanometer-Related Timing Failures," in *Proc. of Intl. Conf. on VLSI Design*, 2004, pp. 597–602.

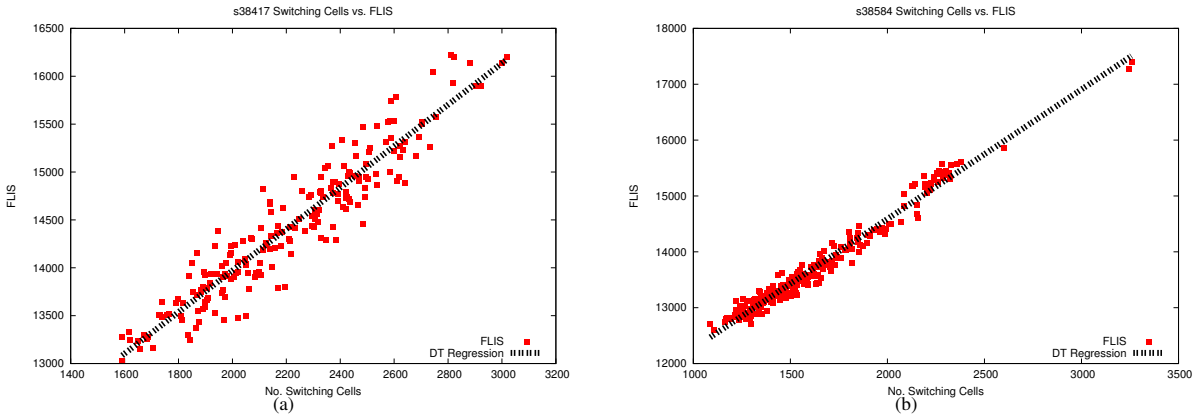

Fig. 5. The linear regression of the FLIS versus number of switching cells for pattern sets of ISCAS'89 benchmarks (a) s38417 and (b) s38584. The FLIS distinctly follows the linear regression as the number of switching cells increases.

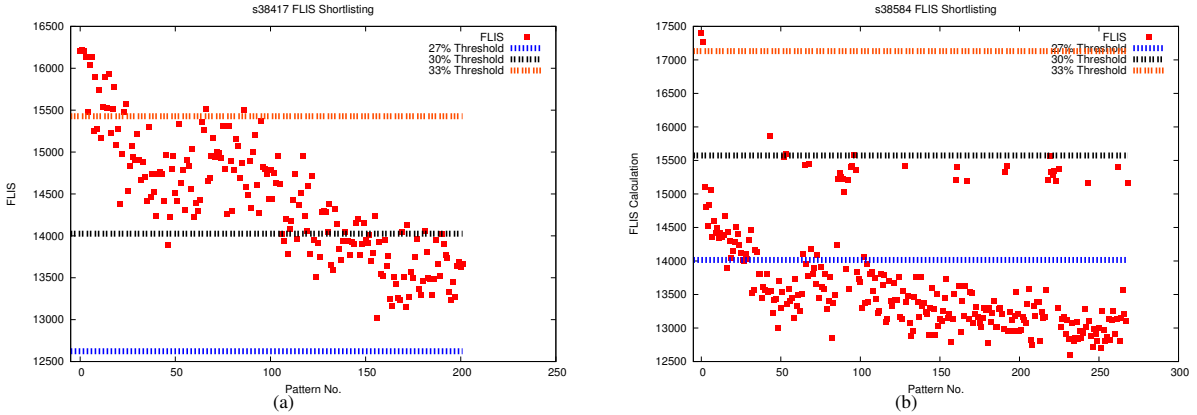

Fig. 6. Shortlisting both (a) s38417 and (b) s38584 for FLIS thresholds of 27%, 30%, and 33%.

TABLE III

LS-TDF FLOW EFFECT ON PATTERN SET WITH 30% THRESHOLD.

Benchmark	Total No. of Cells	TDF Pat. Count	TDF FC %	TDF Ave. Switching	LS-TDF$_1$ Pat. Count	LS-TDF$_2$ Pat. Count	LS-TDF Pat. Count
s1196	370	31	24.51	32	31	0	31
s1238	372	21	23.22	34	21	0	21
s5378	1011	109	68.22	278	109	0	109
s9234	849	158	89.17	198	157	1	158
s13207	2730	135	82.55	592	132	14	146
s15850	2966	108	73.24	617	107	4	111
s35932	7509	28	83.26	2753	0	36	36
s38417	7891	202	97.84	2211	69	388	457
s38584	8472	269	81.06	1618	264	19	283

[7] S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz, and J. Rajski, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," in *Proc. of Intl. Test Conf.*, Oct. 2006, pp. 1–10.

[8] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Supply Voltage Noise Aware ATPG for Transition Delay Faults," in *Proc. of VLSI Test Symposium*, May 2007, pp. 179–186.

[9] X. Wen, Y. Yamashita, S. Kajihara, L. T. Wang, K. K. Saluja, and K. Kinoshita, "Low-Capture-Power Test Generation for Scan Testing," in *Proc. IEEE VLSI Test Symposium*, May 2005.

[10] J. Wang, D. M. H. Walker, A. Majhi, B. Kruseman, G. Gronthoud, L. E. Villagra, P. van de Wiel, and S. Eichenberger, "Power Supply Noise in Delay Testing," in *Proc. of Intl. Test Conf.*, Oct. 2006, pp. 1–10.

[11] X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K. K. Saluja, L. T. Wang, K. S. Abdel-Hafez, and K. Kinoshita, "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing," in *Proc. IEEE VLSI Test Symposium*, May 2006.

[12] "User Manual for Synopsys Toolset Version 2006.06," Synopsys Inc., 2006.

[13] "User Manual for Mentor Graphics Toolset Version 2007.06," Mentor Graphics Corp.

[14] I. Pomeranz, "On the Generation of Scan-based Test Sets with Reachable States for Testing under Functional Operation Conditions," in *Proc. of Design Automation Conf.*, 2004, pp. 928–933.

[15] Y.-C. Lin, F. Lu, K. Yang, and K.-T. Cheng, "Constraint Extraction for psuedo-functional Scan-Based Delay Testing," in *Proc. of Asia and South Pacific Design Automation Conf.*, Jan. 2005, pp. 166–171.

[16] "ISCAS'89 Benchmarks," 1989. [Online]. Available: http://www.fm.vslib.cz/ kes/asic/iscas

[17] *IEEE Std 1364 -2005 IEEE Standard for Verilog Hardware Description Language*, 2006.