Layout-Aware, IR-Drop Tolerant Transition Fault Pattern Generation*

Jeremy Lee, Sumit Narayan, Mike Kapralos, and Mohammad Tehranipoor
ECE Department
University of Connecticut
{jslee,sumitn,mikepk,tehrani}@engr.uconn.edu

***Abstract*— Market and customer demands have continued to push the limits of CMOS performance. At-speed test has become a common method to ensure these high performance chips are being shipped to the customers fault-free. However, at-speed tests have been known to create higher-than-average switching activity, which normally is not accounted for in the design of the power supply network. This potentially creates conditions for additional delay in the chip; causing it to fail during test. In this paper, we propose a pattern compaction technique that considers the layout and gate distribution when generating transition delay fault patterns. The technique focuses on evenly distributing switching activity generated by the patterns across the layout rather than allowing high switching activity to occur in a small area in the chip that could occur with conventional delay fault pattern generation. Due to the relationship between switching activity and IR-drop, the reduction of switching will prevent large IR-drop in high demand regions while still allowing a suitable amount of switching to occur elsewhere on the chip to prevent fault coverage loss. This even distribution of switching on the chip will also result in avoiding hot-spots.**

I. Introduction

Chip performance has constantly been a driving factor of VLSI design. The smaller technologies and higher clock frequencies that contribute to better performance have also created new power management and signal integrity issues that must be addressed during design and test [1]. Maintaining a clean power supply is becoming increasingly difficult in deep sub-micron technologies due to decreasing supply margins and increasing parasitic effects created by smaller wire widths and sharper clock edges. As chip density continues to increase and wire widths continue to shrink with each new technology, there becomes a greater likelyhood of more current being drawn by greater switching activity, increasing both average and instantaneous power consumption, which exacerbates IR-drop and ground bounce.

Although current commercial design tools consider power supply noise, often the power budget only accounts for functional (mission) mode operation. This becomes a significant issue during test since the patterns applied often create higher-than-average switching activity. In an attempt to reduce pattern count, current automatic test pattern generation (ATPG) tools will either randomly fill don't-care states in the pattern to fortuitously detect additional faults that it did not already detect deterministically or fill don't-care bits intelligently to compress the pattern set. For example, the random fill method can initially find many faults, but it also causes much more switching activity and dynamic power consumption than would occur during functional operation of the chip. Transition delay fault (TDF) patterns have been under particular scrutiny since any noise caused by the increased switching activity can create hot-spots and additional delay that was unaccounted for during design, causing the chip to fail during test [2]. There is a need for pattern generation tools that are aware of these power supply noise issues to prevent such overtesting.

A. Related Works

Several works have addressed power supply noise during pattern generation. In [3], a pattern generation technique was proposed to create maximum supply noise to increase the delay along targeted paths. There has also been a pattern post-processing technique to verify patterns that cause excessive IR-drop [4]. Neural network and genetic algorithm based solutions were proposed in [5]. Also, a method of measuring average power called switching cycle average power (SCAP) was used to produce supply noise tolerant patterns [6] [7]. There have also been more precise techniques that perform RLC analysis for pattern generation [8] [9]. However, the extensive analysis required can become time consuming as these networks become more complicated in modern designs.

Additonally, a vector-based compaction solution to reduce overkill and power supply noise induced delay has been proposed in [10] and [11]. The authors developed a vector-dependent power supply noise analysis solution that models the voltage drop based on the layout of the chip. However, simulation of each compacted pattern to estimate IR-drop can result in significant run times. Also, their proposed approach does not consider areas of the chip that may be underutilized.

B. Contribution and Organization

In this paper, we present a novel layout-aware IR-drop tolerant pattern generation method that considers power supply noise issues associated with testing the chip. The method enhances current commercial ATPG tools by taking patterns that are generated based on the post-layout netlist, due to layout synthesis modifications (e.g. clock tree synthesis), and prevents newly generated compacted patterns from creating excessive switching activity, which is so closely related to IR-drop, and underutilization of the entire chip. This allows each pattern to still detect a high percentage of faults while not exceeding the power budget in a centralized area of the chip. We have implemented our proposed method on several ISCAS'89 benchmarks and the results show that coverage is not loss from compaction, pattern count is only minorly affected, and hot-spots are prevented.

The remainder of this paper is organized as follows. Section II presents a discussion on the importance of switching locality on IR-drop. Section III presents a breakdown of the proposed approach, including pattern generation, layout switching

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

978-3-9810801-3-1/DATE08 © 2008 EDAA

profiling, and compaction. The results of our layout-aware compaction are presented in Section IV. Finally, in Section V, we conclude with some closing discussion.

II. Switching Locality

Previous approaches to test power reduction during TDF pattern generation have focused on the quantitative and temporal relation by reducing the switching activity induced by the patterns. While this seems to be an effective approach at reducing power, it also could entail a significant increase in pattern count. As a trivial example, a pattern that induces switching in multiple gates will be modified to only allow a small subset of the switching activity. If these gates are all located near one another, it may be wise to change the pattern to reduce the switching activity. However, if they are evenly distributed across the chip and likely drawing current from different supply pads, the noise induced by one switching gate may have little effect on the power supply network of the other switching gates. Changing the pattern would result in a loss of coverage, requiring a second pattern to recover the faults that were omitted from the first modified pattern. As a result, little is gained in terms of generating less noise on the power supply network.

Reducing the transitions in the chip too much in a layout-unaware approach is not the only potential side-effect of such methods. These approaches can still create the opportunity for hot-spots in the chip to arise. Although the switching activity in the chip is reduced to average functional activity, the potential for all of the switching to occur in a centralized area rather than distributed across the entire chip still remains. This will not only still cause excessive delay due to IR-drop but also potentially damage the chip due to the high current demand.

Therefore, locality of the switching activity can play as much of a role as the number of gates switching and the timing of transitions in determining power supply noise. Suppose there are two simultaneous transitions after applying a test pattern to the chip. The transition that occurs in one corner of the chip may have little impact on the transition occuring in another corner due mainly to the presence of parasitic elements (resistance and capacitance) on supply lines between the two switching gates.

An approach that considers the location of switching activity in relation to both the power supply network and other switching gates would avoid both problems. If the switching activity of a pattern is profiled against the layout during pattern generation, it can be determined if the entire chip is being effectively exercised and no one location is being over-tested. These layout-aware generated patterns can tolerate higher than functional switching activity although no one region is being overly exercised.

III. Pattern Generation Flow

The proposed IR-drop tolerant pattern generation technique is composed of four major steps: *i*) Unfilled TDF Pattern Generation; *ii*) Pattern Transition Monitoring; *iii*) Layout-aware Profiling; and *iv*) Compaction. Unlike conventional pattern generation and compaction, the focus is on producing a pattern set that prevents excessive switching in any one region of the chip in addition to reducing the pattern volume.

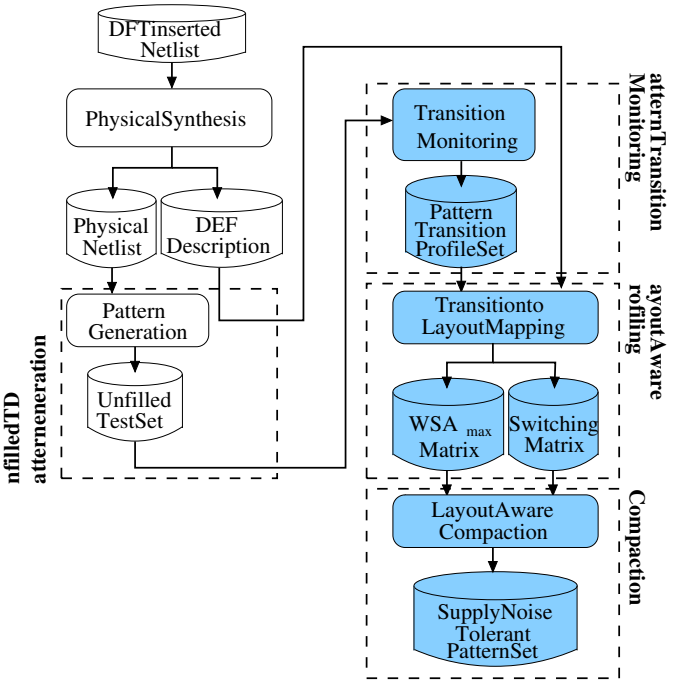

Fig. 1. The IR-drop tolerant pattern generation flow. The flow utilzes the physical layout of the design to monitor where switching activity will occur with each pattern in the transition delay fault pattern set and compact the patterns to prevent over- and under-utilization of the chip.

The flow of this technique is shown in Figure 1 starting from the DFT-inserted netlist, proceeding through physical synthesis, and ending with pattern generation based on layout information. Since the technique relies on knowledge of the physical placement of the gates and the power supply network, unfilled TDF pattern generation cannot occur until after the physical layout is generated. The flow then continues on to pattern transition monitoring, layout-aware profiling, and supply noise tolerant compaction.

A. Unfilled TDF Pattern Generation

ATPG is performed on the post-layout gate netlist rather than DFT-inserted netlist since the layout synthesis may change the gate netlist. Note that the compaction will be dependent on the physical location of the gates. The ATPG engine is set to leave any don't-care states unfilled while it is targeting transition delay faults. This may cause a slight reduction in fault coverage due to missing some faults that may have only been detected with random patterns, but it will allow for more effective compaction later in the flow.

B. Pattern Transition Monitoring

After pattern generation, the patterns are analyzed to determine which gates are switching during transition delay fault pattern application. Although it is possible to use the detected transition delay faults to obtain a broad idea of switching activity, it does not fully characterize all switching created by a TDF pattern; including glitches. As shown in Figure 2, it is possible for multiple fan-out branches to hold a steady value during test and block the observation of many toggling gates.

So, the fault list may give a rough estimate for some patterns but other patterns may have a significant number of transitions that simply cannot be propagated to an observation point due to such masking. Therefore, a simulation-based approach must be used.

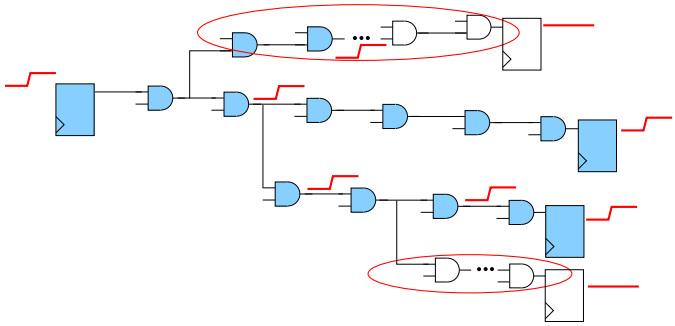

Fig. 2. Transition faults detected by patterns are not a good indication of switching activity. Simulation must be performed to determine all switching activity. Long fan-out paths that have a lot of switching may be blocked and transitions at those fault sites would be undetectable with the given pattern.

Since this is a simulation-based approach, some manner of retrieving the transition information from the simulator is necessary. A Value-Change Dump (VCD) file can be used, but this would be only acceptable for small pattern sets. The Verilog Programming Language Interface (PLI) [12], on the otherhand, allows direct access to internal data while simulating. This eliminates the need for large VCD files and the exhorbitant post-processing time required for such a large VCD file, while requiring little additional overhead by the simulator. Our results have shown that using PLI routines have significantly reduced the CPU run time.

The Verilog PLI subroutines were utilized to monitor which gates switched during the launch-to-capture cycle. A zero-delay simulation is performed and transition arrival times are not considered. Only the final rising or falling transitions are recorded and any glitches during the launch or capture window are ignored. Tracking all transitions due to glitches may provide more accurate results, but also could significantly increase computation time during compaction.

Since fan-out of a switching gate can affect the current drawn, the PLI is also used to determine the number of fan-outs for each switching gate. So, when a transition occurs, both the gate information and the load from the fan-outs will be stored as the pattern transition profile set. With the knowledge of which gates are switching and the load being drawn, the physical location of the switching activity is the last piece that must be determined to perform the compaction.

C. Layout-Aware Profiling

The next phase in the flow needs both the transition results from the pattern transition monitoring step and the location of each of the gates in the layout. Layout information is extracted directly from a DEF (Design Exchange Format) file. Since the layout information is taken from a standard design format, the flow is compatible with current commercial tools and industrial flows.

In addition to the gate placement information, the power supply network is also extracted. The network is then used to create the *maximum weighted switching activity (WSA_{max}) matrix* and *switching matrix*. Both matrices are two-dimensional arrays that can be overlayed on top of the layout that divides it into smaller partitions.

The granularity or number of cells in the matrices can vary depending on the number of power straps/pads/C4 (controlled collapse chip connect) bumps in the design. For very large designs, we suggest each region be located around the power pads (for wirebond packaging) or around the C4 bumps (for flip-chip packages). Figure 3 illustrates, how a physical design will be divided based on the power supply network. In the example, there are four (4) straps vertically across the chip, four (4) straps horizontally across the chip, and power/ground rings around the periphery of the design. Using the straps/rings as midpoints for each cell in the matrices, the chip will then be divided into six (6) columns and six (6) rows for a total of thirty-six (36) cells in both the switching and transition weight matrices. In the case when there are only straps either vertically or horizontally, the direction with the straps will be divided in the same manner as before, while the strapless direction is divided evenly by the same number of cells as the direction with straps. So, if a design has only 4 vertical straps but no horizontal straps, there will be both six (6) columns and rows in the switching matrix and transition weight matrix.

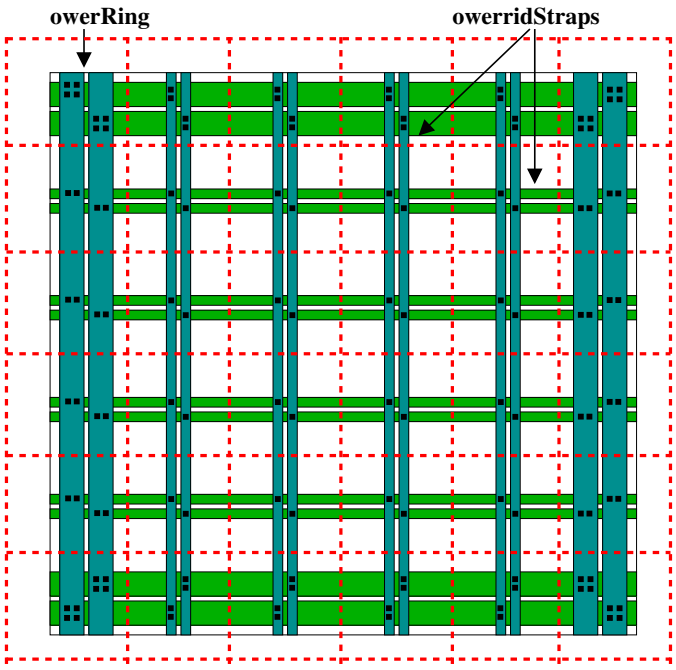

Fig. 3. An example of the power grid straps being used to partition the layout into regions that will have a one-to-one mapping to the WSA_{max} matrix and switching matrix.

The *WSA_{max} matrix* only needs to be created once for each design. Each cell of the matrix will map directly to a region in the layout and store the maximum switching weight for that region. The maximum weighted switching activity ($WSAmax_{ij}$) for a region is calculated with Equation 1.

$$WSAmax_{ij} = \sum_{k} (\tau_k + \phi_k f_k) \quad (1)$$

Each gate, k, in cell (i, j) of the WSA_{max} matrix will be dependent on the weight of a switching gate, τ_k, the number of fan-out of each gate, f_k, and the fan-out load weight, ϕ_k. Since each cell stores the total weight of all the gates in a particular region switching, it will be possible to determine a switching threshold for compaction that is closer to the average functional switching activity for the region.

With the knowledge of both the physical location of all the gates in the layout and which gates are switching, including the fan-out load, a profile of the switching behavior can be generated. This profile for each pattern is stored in the *switching matrix*. The rising and falling transitions that were recorded during transition monitoring are matched to a physical location in the layout and mapped to the appropriate cell of the matrix. The transitions and fan-outs in each region of the layout are weighted, summed, and stored in the respective cell of the switching matrix. The total weight of the transitions occuring at index (i, j) can be summarized by Equation 2.

$$WSA_{ij} = \sum_k D_k(\tau_k + \phi_k f_k),$$
$$D_k = \begin{cases} 1, & \text{Transition Occurs} \\ 0, & \text{No Transition} \end{cases} \quad (2)$$

The equation is similar to Equation 1 except variable D_k is added to account for rising and falling transitions, that are caused during application of the unfilled TDF patterns and to exclude any steady-state signals.

After all switching gates and fan-out loads have been summed and mapped to the switching matrix, a picture of the current demand for each pattern can be realized. Figure 4 shows a hypothetical example of a pattern that has passed through the transition monitoring and layout-aware transition profiling stages of the IR-drop tolerant pattern generation flow. This switching matrix represents the switching weights of a hypothetical single pattern during the launch-to-capture cycle. Each of these matrices will be used to determine those patterns that can be compacted in the following step of the IR-drop tolerant flow.

SampleSwitchingMatri

00	01	02	03	04	05
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Fig. 4. A hypothetical pattern with X-filling has been monitored during simulation and mapped into the switching matrix for any transitions occuring during the launch-to-capture cycle.

D. Compaction

During conventional compaction, test patterns would be compacted as shown in Figure 5. In this example, P1 and P3 either have common care-bits in the same digit positions or care-bits that are unique to one pattern. This will allow P1 and P3 to combine without causing any loss in coverage. After compaction, P'1 becomes the resulting pattern using this simple compaction method. This manner of compaction would continue over the entire pattern set.

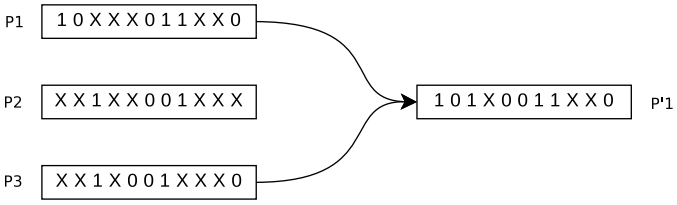

Fig. 5. An example of conventional pattern compaction. Patterns P1 and P3 both share the same care-bits for compaction to be possible.

Although this method may give us the minimum number of patterns for testing, it does not take into consideration the switching activity. The greater the switching activity in a particular region, the greater the current, and the greater the IR-drop. This could result in excessive delays or overheating the circuit-under-test and damage the chip.

Knowing of the location of each gate in the chip/region, after identifying the gates switching, we can infer how a single pattern will affect the current drawn. This data will be used to moderate the amount of IR-drop, which will be used to decide whether to compact the patterns or reject it based on the number of switching gates allowed in a matrix cell. We refer to this number as the *switching threshold*. The switching matrix created earlier will provide the switching information about the pattern needed for IR-drop aware compaction. The compaction can be done in such a way so as to keep the switching gates as evenly distributed as possible without exceeding the switching threshold.

An example of a good match based on switching locality is shown in Figure 6. Patterns P1 and P3 could easily be combined as their matrices indicate that upon combination, the switching activity would still remain reasonably distributed since moderate to high switching activity in each region do not coincide across the chip. Thus, compaction of these patterns would not create excess current flow and would be a good match.

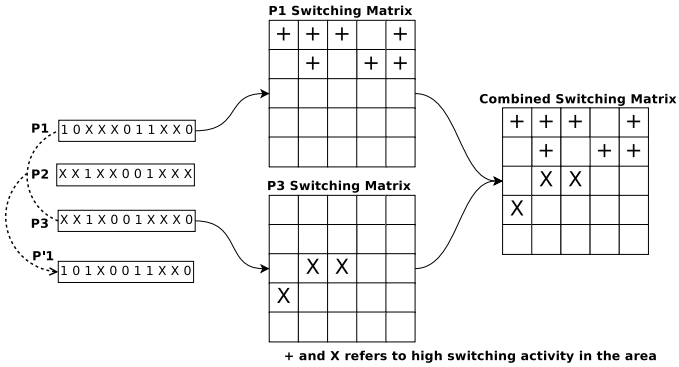

Fig. 6. The result of compacting two patterns that individually create switching activity in different corners of the chip. Upon compaction, the new pattern does not cause the switching activity to rise in one particular region.

The example in Figure 7 shows that P'1, which is obtained

by compacting P1 and P3, can be combined with P4 but causes a lot of localized switching activity. This can be seen in the upper 2 rows of their combined switching matrix. This excessive switching could put too much burden on the power supply grid, drawing much more current than expected during design (may create hot spots) and hence would not be a good choice of compaction for this chip.

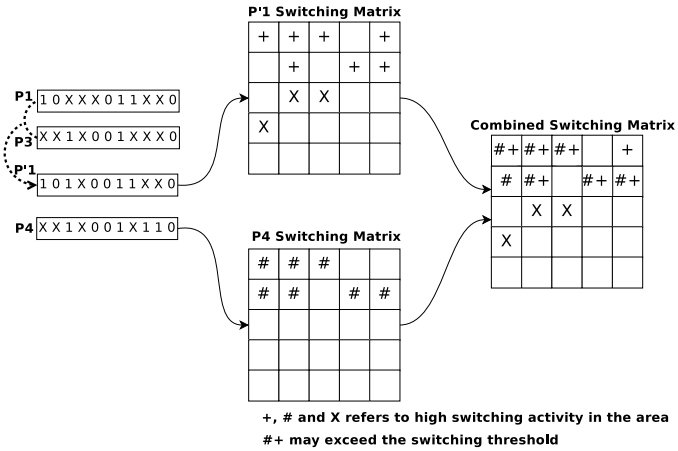

Fig. 7. The result of compacting the result of Figure 6 with another pattern will result in higher-than-average functional switching in the upper half of the design. This compaction result would be rejected.

•**Compaction Algorithm:** A summary of the compaction algorithm used to generate IR-drop tolerant patterns is shown in Figure 8. The algorithm starts with the pattern switching data, i.e., the WSA matrix and switching matrix for each pattern. The compaction algorithm is based on a reverse greedy approach. Starting with the last pattern in the set, it will attempt to perform a trial compaction with the next pattern in the set. This trial compaction will ensure that the next pattern in the set is a good candidate for compaction before checking the switching threshold. If it is a good candidate, the algorithm proceeds to the next phase: switching threshold checking. More complex compaction approaches are possible, but the compaction ratio may only improve by a small percentage, which would not be enough to outweigh the cost of additional run-time.

The *switching threshold* is based on the average functional switching of the design defined by designer. Since the maximum switching activity for each cell, $WSAmax_{ij}$, is known from the WSA$_{max}$ matrix, an appropriate threshold can be obtained by using a percentage of the highest maximum. This maximum can be assumed to be a safe estimate for all regions since the power supply network is generally uniformly designed based on the region with the highest current demand. As two patterns are compacted, the sum of *unique* transitions are added together to create a new switching matrix for the compacted pattern. Each cell of the layout matrix of the compacted pattern must meet one of two criteria. First, compacted patterns cannot cause any cell of the switching matrix to exceed the switching threshold of the chip. If the compacted pattern cannot meet the first criteria due to one pattern alone exceeding the threshold in a cell, then as long as the offending cell does not experience an increase in switching

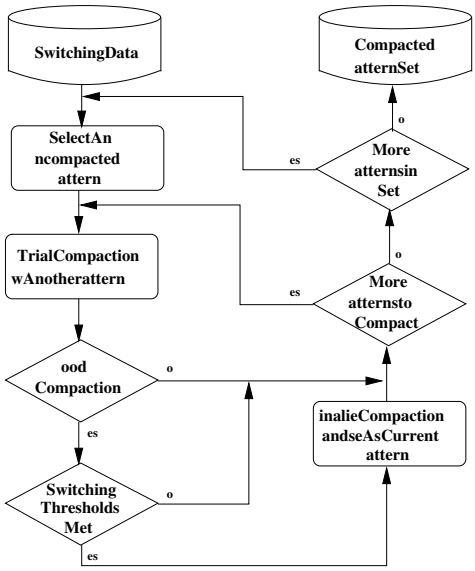

Fig. 8. A flow diagram of the layout-aware compaction algorithm used to compact the initial pattern set into a new pattern set that is tolerant of IR-drop by using the switching activity as a metric.

activity after compaction, the newly compacted pattern passes.

If either the patterns are not good candidates for compaction or the new pattern does not meet the threshold requirements, the pattern is restored and compaction continues with the next pattern in the set. Trial compaction will continue with the remainder of the pattern set. Any patterns that successfully pass trial compaction and meet the threshold criteria are finalized and used as the new current pattern. Once the entire pattern set has been exhausted, the compacted pattern will be placed into the IR-drop tolerant pattern set and the next unfilled TDF pattern will proceed through the same algorithm. Once the entire pattern set has been finalized, the remaining don't-care states in each pattern are filled with zeros since it will not significantly increase the switching activity as much as performing a random fill on the remaining don't-care states.

IV. Experimental Results

The layout-aware, IR-drop tolerant pattern generation flow was implemented on Linux-based x86 architectures with 3-GHz processors and 32 GB of RAM. The gate netlists were synthesized into layout using Synopsys Astro [13], while the unfilled TDF patterns were generated using Synopsys Tetramax [13]. Pattern simulation was performed with Synopsys VCS [13] with the PLI calls implemented in C. The final two phases of the flow, layout-aware profiling and compaction, were also integrated into the Verilog PLI in C/C++.

The flow was tested on the larger ISCAS'89 [14] benchmarks that are listed in the first column of Table I. A power supply network was designed for each benchmark, which included supply rings and straps. Due to the relatively small size of the benchmarks, only vertical straps were used. The number of vertical straps used in each design are listed in the second column and the respective number of cells in the WSA$_{max}$ and switching matrices are shown in the third column of Table I. The initial unfilled TDF pattern count is shown in column 4. Switching thresholds of 20%, 25%

TABLE I

COMPACTION RESULTS OF LAYOUT-AWARE, IR-DROP TOLERANT PATTERN GENERATION

Benchmark	# of Straps	Matrix Size	Unfilled TDF Count	Compaction Ratio		
				20%	25%	30%
s9234	5	49	1068	.472	.497	.506
s13207	5	49	2125	.808	.866	.881
s15850	5	49	2532	.745	.826	.856
s35932	5	49	6010	.995	.995	.995
s38417	6	64	12082	.871	.924	.939
s38584	5	49	10023	.893	.917	.931

TABLE II

COMPACTION RUN TIME

Benchmark	Compaction Time		
	20%	25%	30%
s9234	146s	138s	96s
s13207	184s	194s	168s
s15850	370s	275s	215s
s35932	786s	728s	727s
s38417	4hr 37min	2hr 1min	1hr 40min
s38584	1hr 54min	1hr 49min	1hr 47min

and 30% of the maximum switching for each of the designs were used to generate IR-drop tolerant patterns. The post-compaction pattern count ratios are shown in columns 5–7.

The compaction ratios demonstrate the trend that as the switching threshold increases, the number of patterns after compaction decreases. All benchmarks except s35932 experience this trend. This is likely due to the fact that the unfilled TDF pattern set for s35932 has few care-bits in each pattern allowing for a high compaction ratio with few hot-spot regions.

The compaction time varies based on the switching threshold, the compaction algorithm, and the pattern volume. Columns 2–4 of Table II show the time taken to perform the compaction algorithm on the same ISCAS'89 benchmarks. Pattern sets of the larger benchmarks will obviously take longer to compact due to the size. However, as the switching threshold is increased, compaction time decreases. Since more patterns can be compacted in each pass, subsequent compaction steps will have fewer patterns to check, reducing compaction time.

A. Effects on Fault Coverage

Fault coverage of the proposed pattern generation flow is not significantly affected. Table III shows the results of conventional random-fill TDF ATPG in column 2, the unfilled TDF ATPG used as the initial pattern set in column 3, and the fault simulation results of the IR-drop tolerant patterns in columns 4–6.

When comparing the results of the fault simulation of the IR-drop tolerant patterns with the pre-compaction pattern set, fault coverage only slightly increases due to the zero-fill that is performed post-compaction and some additional faults that are now observable due to the compaction itself. However, for most of the benchmarks, there is not a significant increase of fault coverage from the pre-compaction pattern set to the post-compaction pattern sets. This can imply that the switching activity was not increased since we are not randomly detecting many new faults.

TABLE III

COMPARISON OF RANDOM-FILL TDF PATTERN AND LAYOUT-AWARE IR-DROP TOLERANT PATTERN FAULT COVERAGE

Benchmark	Random TDF Fault Cov.(%)	Pre-Compact. Fault Cov.(%)	Post-Compact. Coverage (%)		
			20%	25%	30%
s9234	86.20	84.70	85.78	85.78	85.78
s13207	78.69	69.80	70.93	70.92	71.01
s15850	70.45	68.80	69.69	69.61	69.61
s35932	81.96	81.04	81.23	81.27	81.26
s38417	97.27	92.60	94.19	94.18	94.21
s38584	79.11	78.30	78.63	78.65	78.67

V. CONCLUSION

We have presented a novel layout-aware, IR-drop tolerant transition delay fault pattern generation flow. This technique targets even distribution of switching activity across the entire design using a layout-aware compaction technique. Otherwise, switching activity could occur in hot-spots and potentially damage the chip during test. Also, the activity is distributed appropriately for the designed power supply network, preventing delay failures due to IR-drop. Fault coverage is also not significantly changed after compaction and pattern generation can be easily integrated with commercial and industrial flows.

REFERENCES

[1] "International Technology Roadmap for Semiconductors (ITRS)," 2005.

[2] 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.

[3] A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Pattern Generation for Delay Testing and Dynamic Timing Analysis Considering Power-Supply Noise Effects," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 3, pp. 416–425, Mar. 2001.

[4] 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.

[5] E. Liau and D. Landsiedel, "Automatic Worst Case Pattern Generation Using Neural Networks & Genetic Algorithm for Estimation of Switching Noise on Power Supply Lines in CMOS Circuits," in *Proc. of European Test Workshop*, 2003, pp. 105–110.

[6] 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.

[7] ——, "Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design," in *Proc. of Design Automation Conference*, June 2007, pp. 533–538.

[8] S. Zhao, K. Roy, and C.-K. Koh, "Estimation of Inductive and Resistive Switching Noise on Power Supply Network in Deep Sub-micron CMOS Circuits," pp. 65–72, Sept. 2000.

[9] C. Tirumurti, S. Kundu, S. Sur-Kolay, and Y.-S. Chang, "A Modeling Approach for Addressing Power Supply Switching Noise Related Failures of Integrated Circuits," in *Proc. of Design, Automation and Test in Europe Conf.*, Feb. 2004, pp. 1078–1083.

[10] J. Wang, Z. Yue, X. Lu, W. Qiu, W. Shi, and D. M. H. Walker, "A Vector-based Approach for Power Supply Noise Analysis in Test Compaction," in *Proc. of Intl. Test Conf.*, Nov. 2005.

[11] 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 Testin," in *Proc. of Intl. Test Conf.*, Oct. 2006, pp. 1–10.

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

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

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