Copyright © 2008 American Scientific Publishers
All rights reserved
Printed in the United States of America

Journal of
Low Power Electronics
Vol. 4, 1–12, 2008

Layout-Aware Transition-Delay Fault Pattern Generation with Evenly Distributed Switching Activity

Jeremy Lee and Mohammad Tehranipoor*

ECE Department, University of Connecticut, Storrs, CT 06269, USA

(Received: 11 June 2008; Accepted: 9 October 2008)

As chip integration continues to increase and technology scaling is forcing the operating voltage to decrease, modern designs have become more susceptible to supply voltage noise. However, even with a well designed power distribution network, modern at-speed test pattern generation techniques do not consider the maximum current throughput the network will be able to provide. As a result, conventional transition delay fault pattern generation tends to create a number of patterns that cause higher-than-average functional switching, which may cause timing and/or functional failures during test. In this paper, we propose a flow that incorporates the layout information and the locality of the switching activity during pattern generation to provide insight into the amount of tolerable switching. This will prevent both IR-drop related hot-spots and under-utilization of the chip since the switching activity can be evenly spread across the design. The results presented in this paper show significant improvement over our previous flow without negatively impacting fault coverage and pattern count.

Keywords: IR-Drop, Layout-Aware, Pattern Generation, Low-Switching, TDF.

1. INTRODUCTION

The complexity of modern designs continues to increase significantly as the feature size continues to shrink due to advances in process technologies. In order to reduce power consumption in these complex devices, the supply voltage has also been scaled appropriately, which make the circuit less tolerable to supply noise that can cause performance degradation or even logic failure. In order to counter the reduced noise immunity, designers must create robust power distribution networks (PDN), which are expected, in practice, to handle approximately 15–20% switching activity in the chip.[1] Designing PDNs for today's technology have become a significant portion of the die area and is not a trivial task.

In addition to decreasing supply voltage, process variations, and parasitic effects are also becoming more apparent. Each of these variables are increasing the probability of timing-related defects, making high-quality, at-speed tests a necessity. Transition delay fault (TDF) patterns have become widely used in practice to detect these timing-related failures and ensure the design correctly functions at the specified frequency.[2–5]

*Author to whom correspondence should be addressed.
Email: tehrani@engr.uconn.edu

Current ATPGs attempt to achieve the highest TDF test coverage while maintaining an adequate pattern volume. To accomplish this, unfilled (don't-care) bits in the pattern are either randomly filled to increase coverage and fortuitously detect unmodeled faults or filled based on a compression algorithm to decrease pattern volume. However, since current ATPGs often ignore timing- and power-related issues, both techniques are known to increase the switching activity beyond the expected average functional switching.[6,7] This will potentially generate IR-drop hot-spots from excessive supply noise, leading to chip failure during test, impacting yield loss since the chip could still function correctly in the field.

Chip designers cannot be expected to design PDNs that would be able to handle at-speed test, especially considering the large area overhead already needed for functional operation. If required to design the PDN for at-speed test, an even larger portion of each metal layer would be dedicated to the PDN making routing congestion almost intractable. Also, since the high switching in test mode will not likely occur during functional operation, there would be a lot of wasted area.

There is a clear need for low-switching TDF pattern generation methods that will prevent patterns from exceeding the power demand of the distribution network of the chip during test. However, it is possible to under-utilize the chip with the low-switching patterns since modern designs often have multiple supply pins, which in the case of flip-chip designs can count into the hundreds. Current layout-unaware low switching pattern generation techniques may isolate switching activity to a small portion of the chip, leaving other areas quiescent while it could actually tolerate additional switching due to the existing additional power supplies.

1.1. Related Works

Previous approaches to reduce power during test have focused on reducing the switching activity during TDF pattern application. The power consumed during test depends on the switching activity, which is generally characterized by the number of cells switching or weighted switching activity (WSA).[8–10]

A pattern post-processing technique (i.e., pattern validation) to verify TDF patterns that cause excessive IR-drop was proposed in Ref. [7]. 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 to produce supply noise tolerant TDF patterns.[11, 12] 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 Refs. [13] and [14]. 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 attempts to reduce the Hamming distance between the initialized, launched, and captured patterns during TDF testing, which would ideally reduce switching activity on the chip.[15, 16] In Ref. [17], the authors propose a low-capture-power (LCP) X-filling method for assigning 0's and 1's 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 Ref. [18] 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. In Ref. [19], the authors modify fully specified at-speed test patterns to reduce switching activity in the circuit. An additional approach has proposed gating the clock to the scan flip-flops and limiting which chain segments capture.[20]

Most previously proposed power reduction methods have focused on power during shift operation either through pattern modification or design changes.[21–24] However, in this work, we only focus on launch-to-capture power during TDF test since a higher percentage of switching activity occurs during the launch-to-capture window of TDF pattern application. Additionally, the capture-to-scan window can be extended with dead-cycles to reduce the potential impact of IR-drop before test mode resumes.

In general, most of the previously proposed methods are layout unaware and, as a result, the approaches that propose low-power pattern sets could still lead to IR-drop hot-spots and under utilization during test.

1.2. Contribution and Paper Organization

In this paper, we propose a transition delay fault pattern generation procedure that considers the potential excessive power supply noise that can arise during test. The proposed procedure:

- monitors the switching gates created by all patterns;
- localizes the switching activity to a particular location in the layout;
- and compacts the patterns using this information to prevent both IR-drop hot-spots and underutilization during at-speed test.

This will allow the pattern set to still detect a high percentage of faults without exceeding the power budget set by the designer during test. Also, the procedure is easily integrated into existing DFT flows.

In addition to the work proposed in Ref. [25], we have extended this work to:

- implement and compare multiple compaction algorithms;
- improve the threshold checking algorithm to significantly reduce the procedure run time;
- present more detailed experimental results;
- and propose a modified flow for large pattern sets that are indicative of industrial designs.

The performance improvement and modified flow makes applying the proposed procedure towards industrial designs more feasible.

The paper is organized as follows. In Section 2, we introduce the importance of considering the layout of the chip during pattern generation. We present our layout-aware pattern generation flow in Section 3. Section 4 discusses our results and compares them to our previous work from Ref. [25]. Finally, we make some concluding remarks in Section 5.

2. SWITCHING LOCALITY

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 occurring in another corner due mainly to the presence of parasitic elements (resistance and capacitance) on supply lines between the two switching gates likely drawing a majority of their current from the nearest power pads. Traditional approaches generally consider the *global switching activity*, which considers switching activity in chip as equally drawing from all supplies regardless of the proximity to the power supply. However, with current and future technologies, switching activity on opposite corners or even a few hundred microns away, can be considered isolated from one another depending on the PDN. As a result, the *local switching activity* or switching activity in the immediate vicinity becomes more significant.

Previous approaches to power reduction during TDF pattern generation have focused on the quantitative and temporal relation by reducing the overall chip switching activity induced by the patterns. These approaches have reduced the *global* switching activity. While these seem to be effective approaches at reducing power, they 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 power supply pads, the noise induced by one switching gate may have little effect on the power supply network of the other switching gates. As a result, the *local* switching activity would be very low. 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 allow IR-drop hot-spots in the chip to arise. Although the switching activity in the chip has been reduced to average functional activity (low global switching), the potential for all of the switching to occur in a centralized area rather than distributed across the entire chip still remains (high local switching). This will not only still cause excessive delay due to IR-drop but also potentially damage the chip due to the high current demand.

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.

3. OVERVIEW OF PATTERN GENERATION FLOW

The proposed pattern generation flow can be divided into four distinct operations: (i) Unfilled TDF Pattern Generation; (ii) Pattern Transition Monitoring; (iii) Layout-aware Profiling; and (iv) Compaction. Each of these steps, including the physical design information needed, are shown in Figure 1. The overall flow is similar to our solution proposed in Ref. [25]. However, further optimizations and new threshold checking mechanism during compaction have been added.

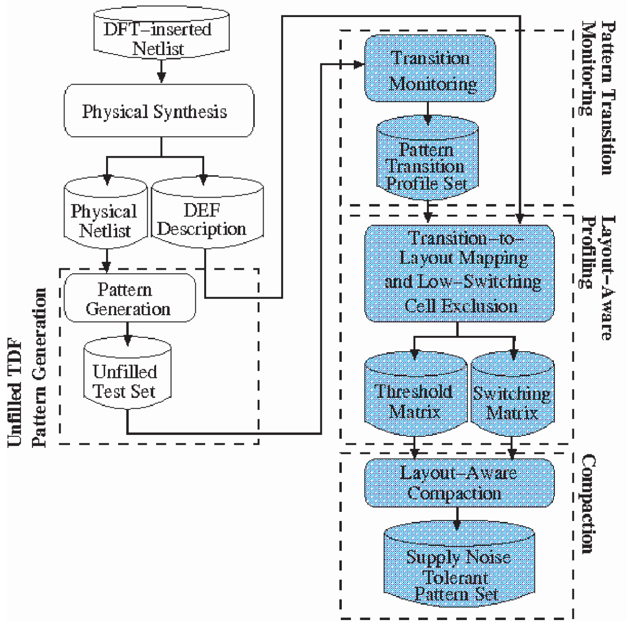


Fig. 1. The IR-drop tolerant pattern generation flow. The flow utilizes the physical layout of the design to monitor where switching activity will occur with each pattern in the transition delay fault pattern set and compact the patterns to prevent hot-spots and under-utilization of the chip.

3.1. Unfilled TDF Pattern Generation

ATPG is performed on the post-layout gate netlist to create an initial pattern for TDF detection. A post-layout netlist is used since layout synthesis may change the gate netlist. Note that the compaction will be dependent on the physical location of the gates since it considers switching locality. In order to improve compaction later in the flow, the ATPG engine is set to leave don't-care states unfilled.

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 incidentally with random patterns, but it will allow for more effective compaction later in the flow and patterns created will generally have little to no switching activity except in the logic cones of the detected fault.

3.2. Pattern Transition Monitoring

After pattern generation, the patterns are analyzed to determine which gates are switching during application of the

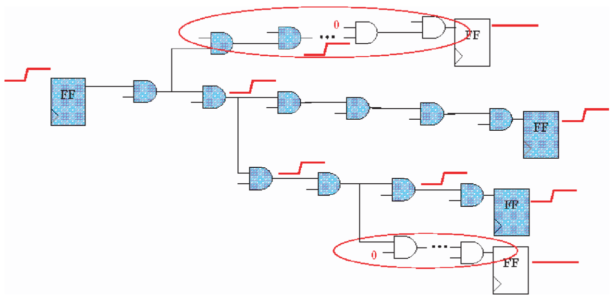


Fig. 2. Transition faults detected by patterns are not a good indication of switching activity. Long fan-out paths that have a lot of switching may be blocked and transitions at those fault sites would be undetectable with the given pattern. Simulation must be performed to determine all switching activity.

transition delay fault patterns. 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, the fault list may provide 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 to determine which gates are switching during the launch-to-capture cycle.

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),[26] on the other hand, allows direct access to internal data while simulating. This eliminates the need for large VCD files and the exorbitant post-processing time required to parse 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. To determine peak and average power, back-annotated simulation based on the physical layout is performed to detect glitches and approximate arrival times.

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.

3.3. Layout-Aware Profiling

To determine where the switching gates are located, layout information is extracted directly from a DEF (Design Exchange Format) file. In addition to the gate placement information, the power supply network is also extracted. The network is then used to create the *switching matrix* and *threshold matrix*. Both matrices are two-dimensional arrays that divide the layout into smaller regions, which will allow us to determine the local switching activity in the circuit.

Region or cell granularity in the matrices can vary depending on the number of power straps/pads/C4 bumps in the design. In this paper, regions refer to the layout while cells refer to the respective region in the switching or threshold matrix. Figure 3 illustrates, how a physical design can 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, which will correspond to the layout being divided into thirty-six (36) regions and a threshold matrix and switching matrix with thirty-six cells. 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 cases 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

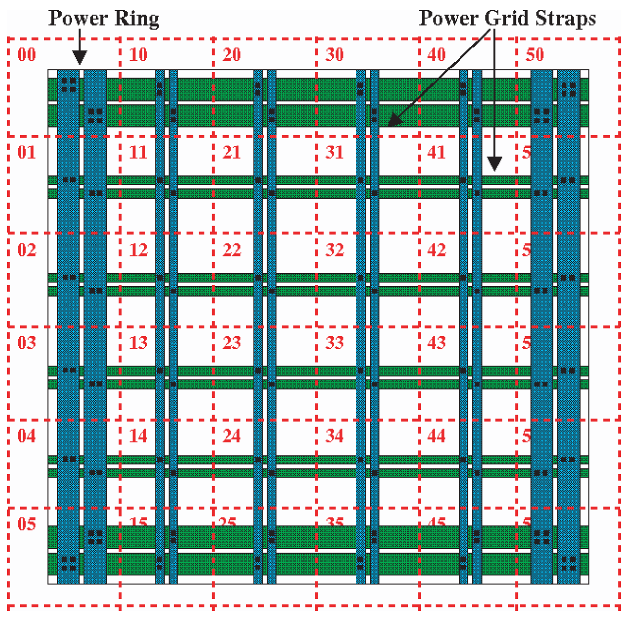


Fig. 3. An example of the power grid straps being used to partition the layout into regions that map to the cells of the switching and threshold matrices.

both six (6) columns and rows in the switching matrix and threshold matrix.

Each pattern will have a respective *switching matrix*, which will map the switching activity monitored during transition monitoring to a region on the layout. The *threshold matrix* only needs to be created once for each design. Again, each cell of the matrix will map directly to a region in the layout and store the switching threshold for that region. Since the number of tolerable switching gates in each region is dependent upon the PDN design, the threshold for both average and peak switching will be dependent upon the PDN design too.

Threshold calculation. The switching threshold for each region of the layout relies heavily on the maximum tolerable current through a region of the PDN. To reduce computational complexity, we focus on determining the maximum allowed peak switching in a cell based on the number of gates in each cell of the threshold matrix and the average functional switching defined by designer. This value will be based on the amount of switching activity required to create significant IR-drop in the cell and cause performance degradation and functional failure, which we assume to be approximately 20% switching activity during functional mode.

To calculate the threshold matrix, the maximum weighted switching activity (WSAmax_{ij}) for each cell is calculated using Eq. (1).

$$\text{WSAmax}_{ij} = \sum_k (\tau_k + \phi_k f_k) \tag{1}$$

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 local switching threshold for compaction that is closer to the average functional switching activity for the region. Assuming a uniform PDN, the cell with the highest WSA_{max} will determine the threshold for all cells of the design since the PDN was designed to maintain good performance given that cell had maximum functional switching. This assumption will also allow us to reduce the number of regions that are checked during compaction. Since some regions will have a WSA_{max} less than the threshold, no matter the applied pattern, the region cannot experience enough switching that will exceed the threshold.

Therefore, these cells can be excluded during compaction. 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 occurring at cell (i, j) can be summarized by Eq. (2).

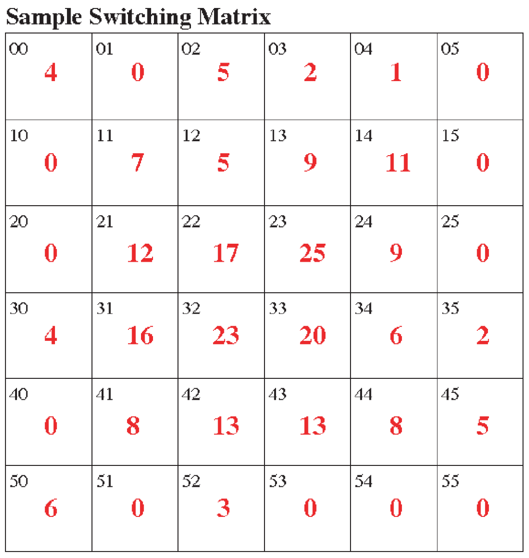
$$\text{WSA}_{ij} = \sum_k D_k(\tau_k + \varphi_k f_k),$$
$$D_k = \begin{cases} 1, & \text{Transition Occurs} \\ 0, & \text{No Transition} \end{cases} \tag{2}$$

Equation (2) is similar to Eq. (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.

3.4. Compaction

Conventional pattern compaction does not consider the location of the switching activity, which could impair chip performance during test. The proposed layout-aware compaction algorithm utilizes the switching and threshold matrices from the previous stage of the flow. This will


Sample Switching Matrix

00 4	01 0	02 5	03 2	04 1	05 0
10 0	11 7	12 5	13 9	14 11	15 0
20 0	21 12	22 17	23 25	24 9	25 0
30 4	31 16	32 23	33 20	34 6	35 2
40 0	41 8	42 13	43 13	44 8	45 5
50 6	51 0	52 3	53 0	54 0	55 0

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

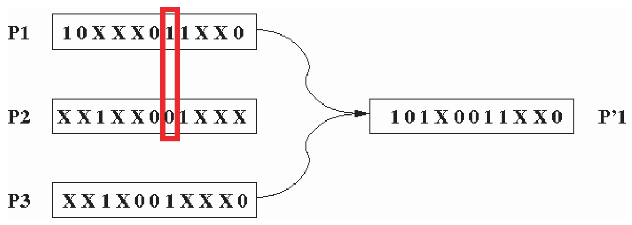


Fig. 5. An example of conventional pattern compaction. Patterns P1 and P3 both share the same care-bits for compaction to be possible.

generate patterns that prevent hot-spots while also utilizing the entire chip.

During conventional compaction, test patterns would generally only be compacted based on the condition of *bit compatibility* as shown in Figure 5. In this example, P1 and P2 cannot be compacted since the 7th bit of the respective patterns conflict. However, P1 and P3 either have common care-bits in the same bit positions or care-bits that is 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.

Although this method may give us the minimum number of patterns for testing, it does not consider the global nor local switching activity, which can cause excessive IR-drop occurring over the circuit due to the number of gates simultaneously switching within the same proximity.

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 *switching compatibility* or number of switching gates allowed in a matrix cell. We refer to this number as the *switching threshold*. The switching matrix created earlier contains 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 compaction based on switching compatibility is shown in Figure 6. Assuming a 25% switching threshold for each cell, patterns P1 and P3 can be combined as their matrices indicate that upon compaction, the switching activity would still remain below threshold since moderate to high switching activity created by the two patterns do not coincide across the chip. Note that upon compaction, since not all switching activity is necessarily unique to each pattern, there may be some common transitions created by the two patterns and, as a result, the combined switching activity is not simply the sum of switching between the two patterns. Since compaction of patterns P1 and P3 would not create excess switching, an

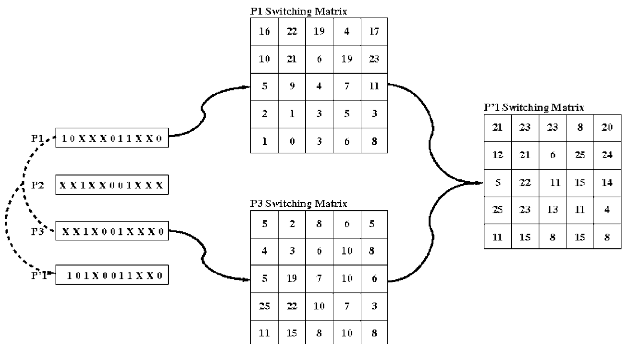


Fig. 6. The result of compacting two patterns that individually create switching activity in different corners of the chip. Upon compaction, the new pattern does not cause the switching activity to exceed the threshold (25%).

IR-drop hot-spot will not be created and the patterns are switching compatible.

The example in Figure 7 shows that P'1, which is obtained by compacting P1 and P3, can be also be compacted with P4 but causes a lot of localized switching activity, exceeding the user-defined 25% threshold. This can be seen in the upper-left region of their combined switching matrix. The 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 would not be switching compatible.

During compaction, it is important to note that when switching matrices are merged, if there is a transition common to both patterns, it is only counted once. So, the lower limit of switching activity after compaction is $SW_T = (SW_1 \cup SW_2) - (SW_1 \cap SW_2)$. There is potential for a small increase in switching activity upon compaction, including glitches. However, the user-defined threshold can be set to compensate for the potential increase, but the likelihood for significant increase will be small since the two patterns either affect different logic cones with little overlap or the same logic cones with few unique care-bits. Also, if two patterns create transitions on the same net but in opposite direction, these patterns will not be switching

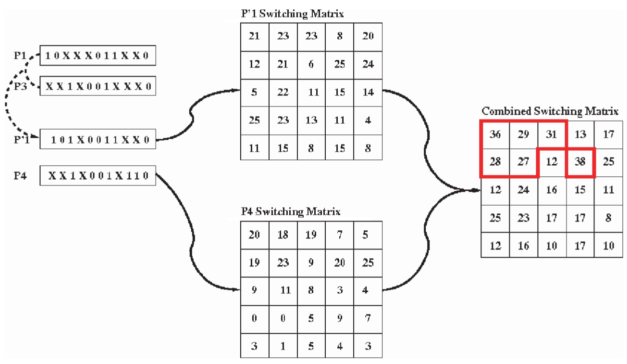


Fig. 7. The result of compaction P'1 and P4 has resulted in higher than threshold switching in the upper half of the design. This compaction result would be rejected.

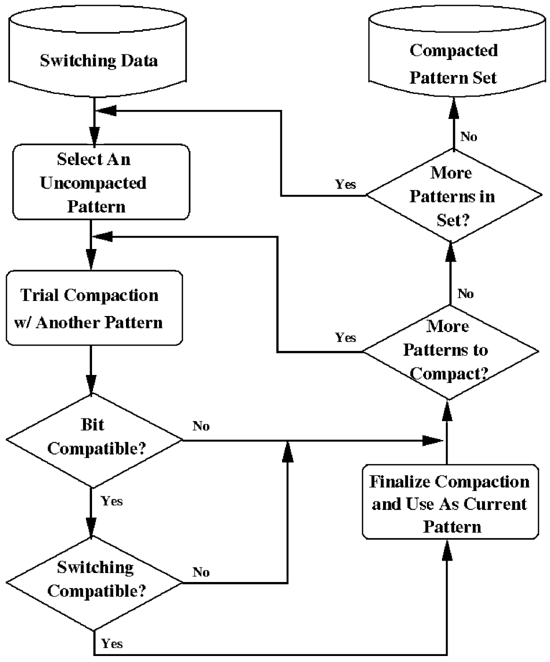

Switching Data
Compacted Pattern Set
Select An Uncompacted Pattern
More Patterns in Set?
Yes
No
Trial Compaction w/ Another Pattern
More Patterns to Compact?
Yes
No
Bit Compatible?
No
Yes
Finalize Compaction and Use As Current Pattern
Switching Compatible?
No
Yes

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. (a) Initial division. (b) Staged compaction and merging.

compatible and more than likely not bit compatible since care-bits have been set to affect the logic cone differently.

3.4.1. Compaction Algorithm

A summary of the compaction algorithm used to generate IR-drop tolerant patterns is shown in Figure 8. In addition to ensuring the patterns can be compacted, it also uses the pattern switching data, i.e., the switching matrix for each pattern and threshold matrix, to ensure the newly compacted pattern does not cause local switching to exceed the switching threshold, preventing hot-spots. At the same time, it may allow higher-than-average functional switching globally across the chip due to switching locality, preventing under-utilization during test.

The algorithm chooses a pattern and then 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 bit compatible, the algorithm proceeds to checking switching compatibility.

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 $\text{WSA}_{\max}$ 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 activity after compaction, the newly compacted pattern passes.

If either the patterns are neither bit nor switching compatible, the pattern original 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.

We have implemented our compaction algorithm in three different ways based on different pattern sorting and the results are shown in Table I. Column 2 shows the number of patterns from unfilled TDF pattern generation. Columns 3–5 show the results of compacting the pattern set in-order, reverse-order, and reverse-order after sorting based on number of don't-care states in the pattern. As the table shows, each of these compaction techniques are able to generate a compacted pattern set with approximately the same compaction ratio. Since the unsorted, reverse-order compaction yields a better result for the larger benchmarks, we will use it during compaction when incorporating the threshold calculation. As will be shown in Section 4, especially for the larger benchmarks, we will be able to approach the ratios shown in Column 4 of Table I, so more complex compaction algorithms may not be needed as they may increase run time significantly.[27]

Table I. Comparison of various compaction techniques w/o considering switching compatibility.

		Compaction technique		
Benchmarks	Uncompacted pattern count	Unsorted in-order (%)	Unsorted reverse order (%)	X-sorted reverse-order (%)
*s*9234	1068	72.3	71.9	70.9
*s*13207	2125	92.9	93.8	93.6
*s*15850	2532	93.7	94.0	93.2
*s*35932	6010	99.4	99.5	99.4
*s*38584	10023	92.7	93.1	92.7
*s*38417	12082	96.0	96.3	96.0

3.4.2. Compaction Complexity

The compaction performance will vary depending on the value set for the threshold. The lower the threshold set, the more patterns that will have to be compacted in later compaction rounds. As a result, if the threshold is set low enough that no two patterns will be switching compatible or no two patterns are bit compatible, the worst possible performance would be O(n2) since a comparison of each pattern with all other patterns would have to be made. However, each round will remove several patterns from the unfilled TDF pattern set and fewer comparisons will need to be made in later rounds. As a result, the average performance will be O(n log n).

3.5. Application to Large Pattern Sets and Industrial Designs

For industrial designs, the base TDF pattern generation step of the proposed flow will likely generate a very large pattern set due to the low coverage each pattern individually generates since such each pattern will likely have a high percentage of bits left unfilled. Even with access to high-powered computers with a lot of memory, which would be available to those in industry, simulating and monitoring the test pattern set in a single pass is not possible.

In order to avoid this computing limitation, dividing the pattern set into subsets and continuing the flow in consecutive passes would eventually compact all patterns together as shown in Figure 9. First the pattern set with M-patterns, where M is very large, is divided into subsets of M/K-patterns each, where K divides the unfilled TDF pattern set based on available computing resources. Each subset then passes through the remainder of the proposed flow (Pattern Transition Monitoring, Layout-Aware Profiling, and Layout-Aware Compaction) creating as many power supply noise tolerant (PSNT) pattern subsets as shown in Figure 9(a). The PSNT subsets then can be combined and passed through the flow again for further layout-aware compaction as shown in Figure 9(b). At most, the flow will have $\lceil \log_2 K \rceil + 1$ stages to compact the entire pattern set.

Although this divide-and-conquer strategy for large pattern sets will make the flow more manageable when faced with limited computing resources, there is a potential reduction in the compaction ratio. Without switching compatibility, when $K = 4$, the compaction ratio for the pattern set of s38417 drops slightly from 96.3% to 95.7%. However, this technique affords the opportunity for parallelism of the layout-aware flow, since each pattern subset can be processed separately. Assuming each compaction stage can be run in parallel when $K = 4$, compaction run time without switching compatibility will decrease by 70%. So, there will be a significant speed increase with little compromise in compaction ratio.

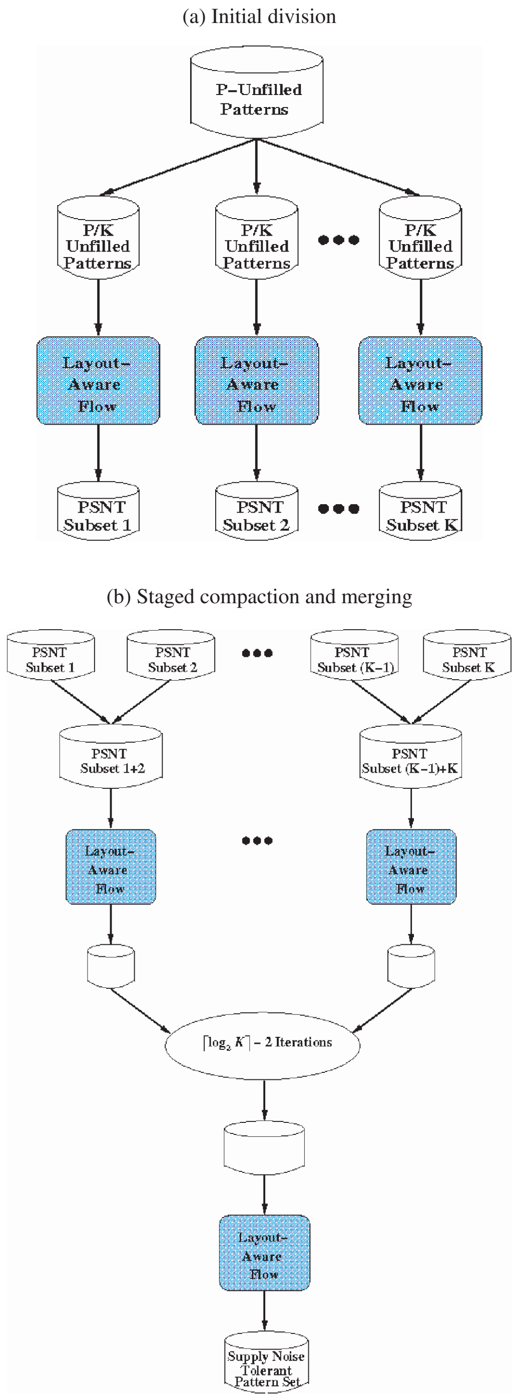


Fig. 9. A divide-and-conquer approach for large pattern sets created during Unfilled TDF Pattern Generation for industrial designs. (a) The initial division of the pattern set in to K-subsets. (b) Compaction and merging of subsets, which will require $\lceil \log_2 K \rceil$ stages to merge into the final supply noise tolerant pattern set. (a) 20% WSA_{max} Threshold, (b) 25% WSA_{max} Threshold, (c) 30% WSA_{max} Threshold.

4. EXPERIMENTAL RESULTS

We ran our proposed IR-drop tolerant, layout-aware pattern generation flow on a Linux-based x86 architecture with 3-GHz processors and 32 GB of RAM. Each of the

Table II. Identified high-switching cells.

Benchmarks	# of straps (n)	Matrix size $(n+2) \times (n+2)$	# of identified high-switching cells 20%	25%	30%
$s9234$	5	49	33	28	26
$s13207$	5	49	35	33	24
$s15850$	5	49	40	39	36
$s35932$	5	49	40	36	35
$s38584$	5	49	49	49	46
$s38417$	6	64	62	57	52

Table IV. CPU run time.

	Compaction time					
	Previous[25]			Current		
Benchmarks	20%	25%	30%	20%	25%	30%
s9234	146 s	138 s	96 s	19 s	11 s	9 s
s13207	184 s	194 s	168 s	31 s	17 s	12 s
s15850	370 s	275 s	215 s	46 s	23 s	15 s
s35932	786 s	728 s	727 s	92 s	72 s	58 s
s38584	1 hr 54 min	1 hr 49 min	1 hr 47 min	928 s	714 s	694 s
s38417	4 hr 37 min	2 hr 1 min	1 hr 40 min	2127 s	949 s	722 s

synthesis steps and unfilled pattern generation were implemented with commercially available tools widely used in industry. The Verilog PLI calls and final two phases of the pattern generation flow, layout-aware profiling and compaction, were all integrated and implemented in C.

The improved flow, including low-switching cell exclusion, was implemented on the largest ISCAS'89 benchmarks, which are listed in Column 1 of Table II. The PDN generated after physical synthesis was used to divide the layout into cells for the switching and threshold matrices. Due to the relatively small size of the benchmarks, only vertical straps were used. The number of straps used is shown in Column 2 along with the number of cells created, in Column 3, with respect to the number of straps. The number of cells kept after low-switching cell exclusion have been listed for 20%, 25%, and 30% thresholds are shown in Columns 4–6. This shows the number of cells that are checked during each compaction round. As the threshold increases, the number of cells with WSA_{max} above the threshold decreases, which partially contributes to run time reduction.

Table III reports the size comparison between a random TDF pattern set and the post-layout-aware compaction set. Columns 2 and 3 show the number of random-filled and unfilled TDF patterns generated, respectively. Column 4 has the average percentage of don't-care bits for each unfilled pattern set. In Columns 5–7, the compaction ratio while considering threshold are shown for 20%, 25%, and 30%. Column 8 shows the compaction ratio without considering switching compatibility. Finally, as a value for comparison, Column 9 reports the size difference between the random and unfilled TDF pattern sets.

As the threshold is increased, more patterns are switching compatible and the compaction ratio increases, reducing the post-compaction pattern set size. In a majority of the cases, a 30% threshold brought the compaction ratio very close to the upper-limit ratio. This can be attributed to a combination of the high average value of don't-care states and relatively low switching created by the unfilled TDF pattern sets. Comparing the post-layout-aware compaction ratios with the size comparison in Column 9 of Table III, for a majority of the benchmarks, even with a threshold of 20%, the ratios are within 5%. So, even though the unfilled pattern set started off much larger than the random TDF pattern set, after compaction, the layout-aware, IR-drop tolerant pattern generation flow does not significantly increase the pattern count in an effort to prevent high localized switching. As a result, pattern inflation is contained while maintaining tolerable IR-drop levels.

Benchmark s9234 represents an exception in terms of reaching the upper-limit ratio and the random TDF ratio. This is likely due to the fairly small size of the circuit and the robust PDN designed for it. Since there were so many straps, each region was divided to include a relatively small number of gates. So, although the thresholds used were 20–30%, due to the existing network, a higher threshold possibly would have been tolerated.

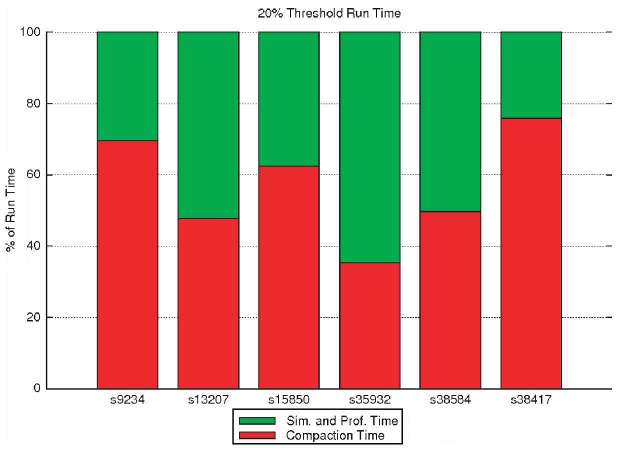
Table IV shows the layout-aware compaction run times for the three different WSA_{max} thresholds of the current method compared to the compaction run time presented in Ref. [25]. While there was improvement in run time, there was little difference between the compaction ratios of Ref. [25] and the results shown in Table III. Columns 2–4 show the compaction run times of our previous algorithm and the current compaction run times as listed

Table III. Comparison of layout-aware, IR-drop tolerant pattern generation and random fill TDF.

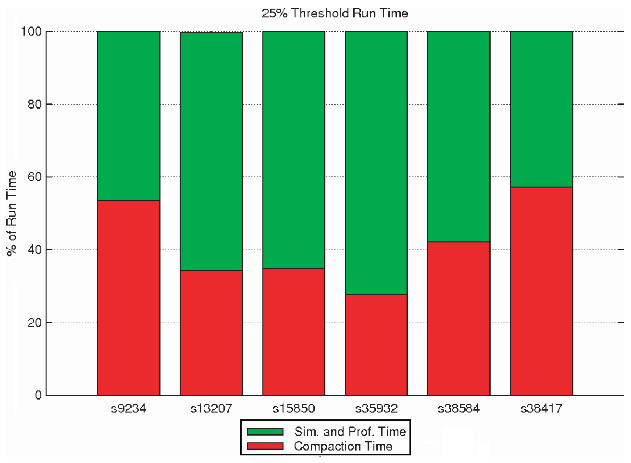
Benchmarks	Random fill TDF pattern count	Unfilled TDF pattern count	% of X's	Compaction ratio 20%	25%	30%	Compaction ratio upper limit	Random to unfilled ratio $(1-\text{R/U})$
s9234	165	1068	85.5	.519	.597	.653	.719	.846
s13207	132	2125	98.3	.908	.923	.926	.938	.938
s15850	118	2532	97.4	.905	.936	.940	.940	.953
s35932	40	6010	99.5	.992	.994	.995	.995	.993
s38584	202	10023	98.9	.927	.930	.930	.931	.980
s38417	276	12082	98.8	.946	.958	.962	.963	.977

in Columns 5–7. When comparing the run times, as the threshold value increases, compaction run time decreases. This can be attributed to more patterns being compacted in earlier iterations, leaving fewer patterns to check later.

(a) 20% WSA$_{max}$ threshold



(b) 25% WSA$_{max}$ threshold



(c) 30% WSA$_{max}$ threshold

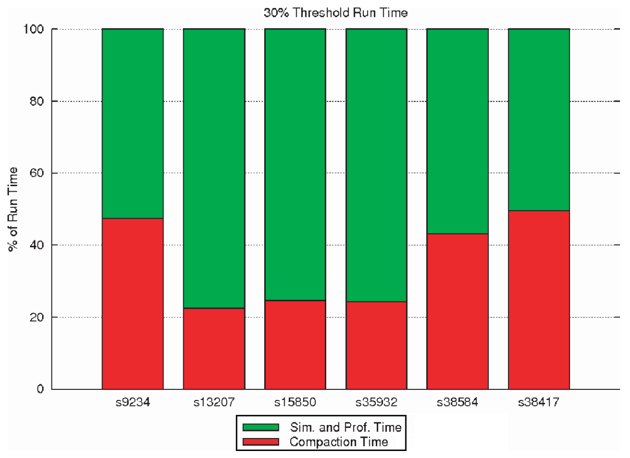


Fig. 10. Bar graphs representing the overall IR-drop tolerant, layout-aware TDF pattern generation flow divided into fractions of time taken for layout-aware compaction and time to perform simulation and profiling.

In addition to low-switching cell exclusion, further program optimizations were made, which significantly benefited compaction time. The worst case compaction run time for s38417 has now been reduced to about 30 minutes from what was over 4 hours earlier. The overall compaction run time depends on the pattern set size, threshold value, and number of cells not excluded from the switching matrix.

Since the layout-aware compaction run time decreases as threshold decreases, compaction becomes a smaller fraction of the overall flow run time since transition monitoring and profiling is essentially a constant in the overall flow run time. Figure 10 breaks down the pattern generation flow run time by compaction time and simulation (transition monitoring) and layout profiling. The percentage of time for layout-aware compaction on s35932 for a 30% threshold in Figure 10(c) is clearly less than the percentage of time for the 20% threshold in Figure 10(a) and the 25% threshold in Figure 10(b).

4.1. Effects on Fault Coverage

Fault coverage of the proposed pattern generation flow is not significantly affected. Table V 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 against the pre-compacted pattern set, fault coverage slightly increases due to the zero-fill of any remaining don't-care bits 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.

Although the fault coverage for the layout-aware pattern sets is lower than the random TDF pattern set, four of the layout-aware sets retain coverage within 1% of the respective random pattern set while a fifth set is within 4% of its respective randomly generated pattern set. This demonstrates the utility of the layout-aware flow in both

Table V. Comparison of random-fill TDF pattern and layout-aware IR-drop tolerant pattern fault coverage.

			Post-compact. fault coverage (%)		
Benchmark	Random TDF fault cov. (%)	Pre-compact. fault cov. (%)	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
s38584	79.11	78.30	78.63	78.65	78.67
s38417	97.27	92.60	94.19	94.18	94.21

obtaining adequate fault coverage while also maintaining IR-drop within acceptable levels. However, s13207 shows that the coverage obtained by the post-compacted set can only reach a little higher than the coverage of the pre-compacted set.

5. CONCLUSIONS

We have introduced our novel layout-aware, IR-drop tolerant transition delay fault pattern generation flow. The flow targets even distribution of switching activity across the entire design to avoid IR-drop hot-spots and under-utilization during test. We have been able to greatly improve the performance of the flow by targeting only those regions that have a possibility of exceeding the user-defined WSA threshold, while safely ignoring those regions that fall below the threshold. The flow is flexible and can be easily integrated in to existing DFT flows, which eases adoption.

The results show that the flow is able to generate pattern sets with pattern counts close to that of conventional TDF pattern generation with random filling. Also, the fault coverage of the layout-aware, IR-drop tolerant sets approaches that of the random-filled sets. The impact on pattern generation run time is flexible and dependent upon the user-defined threshold.

Future directions for this work include considering ground bounce (L di/dt effects) as inductance becomes a more significant issue in the PDNs of new technologies. Also, application to larger benchmarks and industrial-sized designs will be investigated. Since the presented flow relies heavily on don't-care bits, it is not compatible with compression tools that make use of those care bits to reduce pattern count. Further research must be done to determine appropriate use of the flow when compression techniques are utilized in the design during test.

Acknowledgments: This work was supported in part by Semiconductor Research Corporation under contracts No. 1455 and 1587 and National Science Foundation grant CCF-0811632.

References

1. Cadence Design Systems Inc., Power analysis statistical mode. User Manual for SoC Encounter **(2007)**.
2. J. Savir, Skewed-load transition test: Part I, calculus. *Proc. of Intl. Test Conference* **(1992)**, pp. 705–713.
3. J. Savir and S. Patil, On broad-side delay test. *Proc. of VLSI Test Symposium* **(1994)**, pp. 284–290.
4. 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. *Proc. of Intl. Test Conf.*, October **(2002)**, pp. 1120–1129.
5. X. Lin, R. Press, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli, High-frequency, at-speed scan testing. *IEEE Design and Test of Computers* 20, 17 **(2003)**.
6. 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. *Proc. of Intl. Test Conf.*, September **(2003)**, pp. 1098–1104.
7. A. Kokrady and C. P. Ravikumar, Fast, layout-aware validation of test-vectors for nanometer-related timing failures. *Proc. of Intl. Conf. on VLSI Design* **(2004)**, pp. 597–602.
8. P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, A test vector inhibiting technique for low energy BIST design. *Proc. of VLSI Test Symposium* **(1999)**, p. 407.
9. P. Girard, Survey of low-power testing of VLSI circuits. *IEEE Design and Test of Computers* 19, 82 **(2002)**.
10. W. Li, S. M. Reddy, and I. Pomeranz, On reducing peak current and power during test. *IEEE Computer Society Annual Symposium on VLSI: New Frontiers in VLSI Design (ISVLSI'05)* **(2005)**, pp. 156–161.
11. N. Ahmed, M. Tehranipoor, and V. Jayaram, Supply voltage noise aware ATPG for transition delay faults. *Proc. of VLSI Test Symposium*, May **(2007)**, pp. 179–186.
12. N. Ahmed, M. Tehranipoor, and V. Jayaram, Transition delay fault test pattern generation considering supply voltage noise in a SOC design. *Proc. of Design Automation Conf.*, June **(2007)**, pp. 533–538.
13. J. Wang, X. Lu, W. Qiu, Z. Yue, S. Fancler, W. Shi, and D. M. H. Walker, Static compaction of delay tests considering power supply noise. *Proc. of VLSI Test Symposium*, May **(2005)**, pp. 235–240.
14. 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. *Proc. of Intl. Test Conf.*, November **(2005)**.
15. 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. *Proc. of Intl. Test Conf.*, October **(2006)**, pp. 1–10.
16. S. Remersaro, X. Lin, S. M. Reddy, I. Pomeranz, and J. Rajski, Scan-based tests with low switching activity. *IEEE Design and Test* 24, 268 **(2007)**.
17. X. Wen, Y. Yamashita, S. Kajihara, L. T. Wang, K. K. Saluja, and K. Kinoshita, Low-capture-power test generation for scan testing. *Proc. IEEE VLSI Test Symposium*, May **(2005)**.
18. 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. *Proc. IEEE VLSI Test Symposium*, May **(2006)**.
19. X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L.-T. Wang, A novel scheme to reduce power supply noise for high-quality at-speed scan testing. *Proc. of Intl. Test Conf. (ITC'07)* **(2007)**, pp. 1–10.
20. P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, Scan architecture with mutually exclusive scan segment activation for shift and capture power reduction. *IEEE Trans. on Computer-Aided Design* 23, 1142 **(2004)**.
21. R. Sankaralingam, R. R. Oruganti, and N. A. Touba, Static compaction techniques to control scan vector power dissipation. *Proc. of VLSI Test Symopsium (VTS'00)* **(2000)**, pp. 35–40.
22. K. M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Lewis, Minimizing power consumption in scan testing: Pattern generation and DFT techniques. *Proc. of Intl. Test Conf. (ITC'04)* **(2004)**, pp. 355–364.
23. L. Whetsel, Adapting scan architectures for low power operation. *Proc. of Intl. Test Conf. (ITC'00)* **(2000)**, pp. 863–872.
24. M. Elshoukry, M. Tehranipoor, and C. P. Ravikumar, A critical-path-aware partial gating approach for test power reduction. *ACM Trans. on Design Automation of Electronic Systems* 12, 17 **(2007)**.
25. J. Lee, S. Narayan, M. Kapralos, and M. Tehranipoor, Layout-aware, IR-drop tolerant transition fault pattern generation. *Proc. of Design, Automation and Test in Europe (DATE-2008)*, March **(2008)**.
26. *IEEE Std 1364-2005 IEEE Standard for Verilog Hardware Description Language* **(2006)**.
27. J. H. Patel and I. Hamzaoglu, Test set compaction algorithms for combinational circuits. *Proc. of Intl. Conf. on Computer-Aided Design (ICCAD'98)* **(1998)**, pp. 283–289.

Jeremy Lee

Jeremy Lee *received B.S. degree in Computer Engineering from the University of Maryland Baltimore County (UMBC), Baltimore in 2003. He is currently seeking his Ph.D. from the University of Connecticut, Storrs. His current research interests include computer-aided design and test and on-chip environmental variations due to IR-drop and crosstalk effects.*

Mohammad Tehranipoor

Mohammad Tehranipoor *received the B.Sc. degree from Amirkabir University of Technology (Tehran Polytechnic University), Tehran, Iran, in 1997, the M.Sc. degree from the University of Tehran, Tehran, in 2000, and the Ph.D. degree from the University of Texas at Dallas, Richardson, in 2004, all in electrical engineering. Dr. Mohammad Tehranipoor is currently an Assistant Professor of Electrical and Computer Engineering at the University of Connecticut. He was also on the faculty of CSEE department at University of Maryland Baltimore from 2004 to 2006. His current research projects include: computer-aided design and test for CMOS VLSI designs and emerging nanoscale devices, design-for-testability, at-speed test, secure design and IC trust. He is a recipient of several major grants from NSF, SRC and semiconductor companies. Dr. Tehranipoor has published over 85 journal articles and refereed conference papers in the area of VLSI design and test. He has published two books in addition to two book chapters. He is a recipient of a best paper award at the 2005 VLSI Test Symposium (VTS), best paper award at the 2008 North Atlantic Test Workshop, best paper candidate at the 2006 Design Automation Conference (DAC), and best panel award at VTS'2006. He serves on the program committee of several conferences and workshops. He served as Program Chair of the 2007 IEEE Defect-Based Testing (DBT) workshop. He is currently serving as Co-program Chair of the 2008 International Defect and Fault Tolerance in VLSI Systems (DFT) and Program Chair of the DBT'2008. He has organized a new workshop called IEEE International Workshop on Hardware-Oriented Security and Trust (HOST) and is serving as General Chair of HOST. He is also currently serving as an Associate Editor for JETTA. Dr. Tehranipoor is a Senior Member of the IEEE and Member of ACM and ACM SIGDA.*