

On Power-profiling and Pattern Generation for Power-safe Scan Tests

V.R. Devanathan, C.P. Ravikumar
ASIC, Texas Instruments India Pvt. Ltd.
Bangalore, India - 560 093
{vrd,ravikumar}@ti.com

V. Kamakoti
Department of Computer Science and Engg.
Indian Institute of Technology, Madras, India - 600 036
kama@cs.iitm.ernet.in

Abstract

With increasing use of low cost wire-bond packages for mobile devices, excessive dynamic IR-drop may cause tests to fail on the tester. Identifying and debugging such scan test failures is a very complex and effort-intensive process. A better solution is to generate correct-by-construction “power-safe” patterns. Moreover, with glitch power contributing to a significant component of dynamic power, pattern generation needs to be timing-aware to minimize glitching. In this paper, we propose a timing-based, power and layout-aware pattern generation technique that minimizes both global and localized switching activity. Techniques are also proposed for power-profiling and optimizing an initial pattern set to obtain a power-safe pattern set, with the addition of minimal patterns. The proposed technique also comprehends irregular power grid topologies for constraints on localized switching activity. Experiments on ISCAS benchmark circuits reveal the effectiveness of the proposed scheme.

1. Introduction

Scan-based testing is widely practiced in the industry. It has two distinct phases, namely, the *shift* and the *capture*. In the shift phase, a test pattern is loaded into the scan chain, while simultaneously unloading the response from the previous test pattern. During the capture phase, the device is placed in the functional mode, and response for the test pattern loaded during the shift phase is captured into the flip-flops. Although scan-based test is simple to apply, it creates new challenges in the nanometer technologies. It has been observed that scan test power is significantly higher than the functional test power [1], [2]. Moreover, highly localized peak test power results in IR-drop related failures [1] and can impact the yield. On the other hand, it has been observed that power-aware test could improve reliability and yield of the device [3].

Excessive IR-drop becomes an issue with delay tests, as the launch and capture pulses are applied at-speed [1]. This problem aggravates with the increasing use of low-cost wire-bond packages for mobile devices [4]. Silicon debug of patterns that fail in the tester due to excessive IR-drop is a tough task. Extensive simulations or layout-aware static verification techniques [5] are usually employed to detect IR-drop failures. For the patterns that fail on the tester, two options remain: (a) accept the coverage loss by avoiding the patterns, or (b) regenerate power-safe patterns to recover the lost coverage. The latter is a challenging task, compared to the former. *Pattern scrubbing* techniques have been previously employed in [1] to overcome such failures. But these techniques are ad hoc and do not deterministically reduce excessive IR-drop. Hence, there is an important need to generate correct-by-construction patterns, that do not violate power constraints, to ensure reliable delay tests.

1. Prior Work

Several techniques have been proposed to reduce scan test power

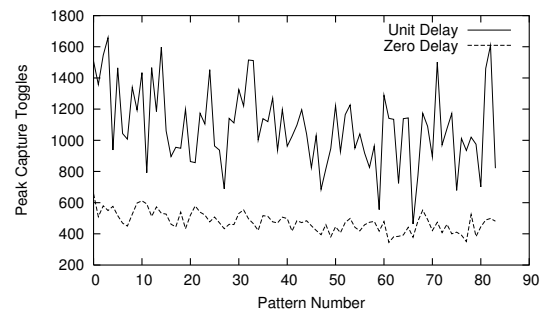


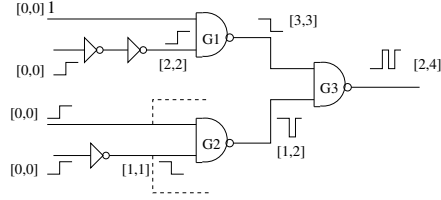
Fig. 1. Power Profile of a pattern set for c2670

[6]. They can be broadly classified as *Scan architecture-based* techniques and *ATPG-based* techniques. Scan architecture-based approaches [2], [6]–[8] include techniques such as scan chain re-ordering, clock-gating, scan chain partitioning, supply-gating, etc. Though these techniques minimize peak shift and capture power, they do not minimize localized peak power that may result in IR-drop failures. A multi duty-scan technique has been proposed in [9] to reduce excessive IR-drop during scan shift operation. This technique uses staggered shift cycles to reduce simultaneous switching activity during shift operation. But, excessive IR-drop during the capture phase is not reduced by this technique. Since the scan architecture is finalized much before the physical layout, scan architecture-based techniques cannot deterministically reduce highly localized switching activity during capture cycle. ATPG-based approaches, on the other hand, include techniques such as low power X-filling, pattern ordering techniques and low power pattern generation schemes [2], [6], [10], [11]. These techniques also minimize peak shift and capture power, but do not reduce highly localized switching activity. Since existing techniques such as multi-duty scan, supply-gating and X-filling are very good for reducing shift power, we do not consider shift power in this paper. Instead we focus on reducing the global and localized switching activity during the capture phase.

2. Role of Timing Information

None of the above low power pattern generation schemes use timing information. Figure 1 shows the peak switching activity using zero and unit delay model, for patterns generated by a commercial ATPG tool for c2670 ISCAS85 benchmark. Other circuits also show similar behavior. The following observations can be made.

- 1) Peak switching activity with unit delay model is significantly higher than that of zero delay model. Similar observations have been made in [12]. The reason for this behavior is that glitch power, due to static and dynamic hazards, contributes to a significant portion of the dynamic power. Figure 2 shows a sample simulation with both static and dynamic hazards with



LEGEND

- Fanout to other gates
- ┌ Rise transition
- └ Fall transition
- Static 1 hazard
- ▬ Dynamic rise hazard
- [a,b] Signal may be unstable during the interval [a, b]
Signal is stable for time $t < a$ and $t > b$

Fig. 2. Sample Scenario with Static and Dynamic Hazards

TABLE I
PEAK SWITCHING PATTERNS FOR C1355

Pattern	Zero-Delay		Unit-Delay	
	Toggle Rank	Toggle Count	Toggle Rank	Toggle Count
P0	1	252	16	564
P1	2	246	120	450
P2	3	238	22	552
P3	81	201	1	621
P4	40	213	2	619
P5	84	200	3	614

unit delay model, assuming all inputs transition at time 0. These hazards are not noticed when zero delay model is used for analysis. It may also be seen that the hazardous output gets propagated till the primary output or the data input of the flip-flop, possibly causing more hazards, thereby increasing the dynamic power significantly. It has been observed in [13] and [14] that reducing glitches (or hazards) can result in as much as 70% reduction in total dynamic power. Moreover, it has been observed in [15] that hazards might result in yield-loss and might cause fault-masking of delay faults.

- 2) There is no correlation between the pattern corresponding to maximum switching activity with unit and zero delay models. Table I shows the patterns, from a commercial ATPG tool, that dissipates high peak power with zero or unit delay model for c1355 ISCAS85 benchmark. The toggle rank of a pattern indicates its position when the patterns are sorted in the descending order of the switching activities. Patterns P0 to P2 are the top 3 patterns for the zero delay model, while patterns P3 to P5 are the top 3 patterns for the unit delay model. Pattern P0, which has the maximum switching activity in zero delay model, ranks 16 in unit delay. Patterns P1 and P2, ranking 2 and 3 with zero delay model, rank poorly in unit delay. Similarly, pattern P3, which has maximum peak switching activity in unit delay ranks poorly at 81 in zero-delay. **Hence, we may conclude that any ATPG scheme that aims to deterministically minimize the peak switching activity needs to be timing aware to correctly estimate and minimize the peak switching activity.**

3. Contribution and Organization of this paper

In this paper, we observe that timing information is crucial for estimating and minimizing peak switching activity. We propose a power and layout-aware, timing-based ATPG scheme that minimizes global and localized switching activity, considering the effects of hazards. We also propose an integrated pattern-optimization flow that

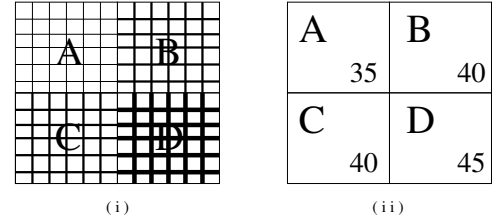


Fig. 3. (i) Locally Regular Globally Irregular Power Grid, (ii) Allowable Localized Switching Activity for Each Region

performs power-profiling, fault-grading and pattern generation, from an initial pattern-set to arrive at an optimal pattern-set that satisfies power constraints with minimal extra patterns.

The organization of this paper is as follows. Section 2 presents the proposed pattern generation scheme. Section 3 describes the proposed integrated pattern optimization flow. Experimental results on ISCAS benchmark circuits are presented in Section 4. Section 5 concludes the paper.

2. Power and Layout-Aware, Timing-based Pattern Generation

1. Overview of the Proposed Framework

1) *Overview of Power Grid Topology*: Typically, the power grid topology for a chip is designed based on the functional power consumption pattern of various parts of the chip. Functional power is not uniformly spread across all regions of the chip. Hence, over-designing the power grid by having uniform grid for the entire die targeting the worst-case scenario results in wasted die area. It is common to use locally regular and globally irregular power grid design, as shown in Figure 3 [16]. In Figure 3(i), region A with low power consumption has thin power straps, regions B and C with moderate power consumption have normal sized power straps, and region D with maximum power consumption has thick power straps. Region A has maximum allowable localized toggle limit of 35, while region D allows up to 45 toggles as the grid for region D has thicker power straps. For SoCs with hard-IPs (or pre-placed cores), the power grid may be non-uniform, and the maximum allowable localized switching activity would vary across grid locations. For irregular power grid design, the allowable maximum switching activity could be stated as a vector, such as $\langle 35, 40, 40, 45 \rangle$, for regions A, B, C and D of Figure 3(ii).

2) *Proposed Layout-Aware Framework*: In the proposed framework, the design is tessellated into coarse regions, based on the physical layout. Each gate in the design is then assigned a region, corresponding to its physical location. The switching activities of all gates within a region are separately stored for each region co-ordinate. Difference in the internal power dissipation of various gates is accounted by scaling the switching activity of each gate with respect to its internal power compared to that of an inverter. To comprehend loading effects, the switching activity at each gate output is scaled based on the fanout. If multiple voltage domains and multiple clock domains are permitted, we can scale the switching activity of a gate by $\alpha x V^2 x g x f$, where α denotes the internal power of the gate with respect to an inverter, V denotes the voltage domain of the gate, g denotes the fanout of the gate and f denotes the capture clock frequency.

Figure 4 shows a sample 3x3 array of regions, with each region assigned a co-ordinate pair. Gates G1 and G2 lie in region (1,1), gates G3, G4 and G5 lie in region (1,2), gates G8 and G9 lie in region (2,1), and so on. The switching activity information for a sample

	1	2	3
1	G1, G2 2	G3, G4, G5 4	G6, G7 1
2	G8, G9 1	G10, G11 1	G12, G13, G14 3
3	G15 0	G16, G17 1	G18, G19 1

Fig. 4. A Sample Region

pattern is also shown in the bottom right corner of each region. The region (1,1) has 2 toggles, region (1,2) has 4 toggles, region (3,1) has 0 toggles, etc.

Constraints for global and regional toggle counts are used to ensure that *for any pattern, the total number of toggles for all regions is lesser than the global toggle count constraint, and number of toggles within each region is lesser than the regional toggle count constraint.* The global toggle constraint ensures that the global peak power is within the limits, while the regional toggle constraint ensures that high localized switching activity, that could potentially lead to IR-drop failure, does not occur.

3) *Proposed Timing-based Framework for Pattern Generation:* A lumped gate delay model is used for performing timing-based hazard analysis. The earliest and the latest expected signal arrival times, obtained by Static Timing Analysis (STA), are stored for each signal. Low values of the timing window size, indicate that paths leading to this gate are well-balanced. If the size of the timing window is high, it means that there are paths with large difference in signal arrival times that could lead to a hazard. For example, Figure 2 shows a case where a static-1 hazard is created at the output of gate G2, due to the difference in input arrival times. Further, the static hazard propagates in the fanout cone to cause more hazards in gate G3. Hence, it could be inferred that the probability of hazard in the output of the gate is proportional to the timing window size.

2. Proposed Pattern Generation Scheme

The reader may refer to [17] for a description of the PODEM algorithm. Although PODEM was originally proposed for combinational circuits, for full scan circuits, we can treat the output of flip-flops as pseudo input and the input of flip-flops as pseudo output and concentrate on the combinational logic cloud that separates the sequential elements. Again, PODEM was originally proposed for generating patterns for stuck-at faults, but we can extend the algorithm for transition delay faults using technique proposed in [18].

If a 0 to 1 slow-to-rise fault is to be detected at fault location s , we need the following: (a) Use PODEM to initialize s to 0. (b) Use PODEM to launch the transition at s through a functional path. (c) Use PODEM to propagate the effect of the fault to an output. PODEM uses the following two procedures to set values on the primary inputs so as to activate the fault and propagate the effect of the fault to a primary output:

- 1) The Backtrace(s) procedure starts from the fault location s and traces a path back to an input I . The value on I is implied a value v that is most likely to meet the *objective* of the algorithm.
- 2) The Objective() procedure generates an objective. For example, if we are dealing with a slow-to-rise transition delay fault at s , then the initial objective will be to set s to 0.

The conventional PODEM algorithm uses randomization to select a path among the cone of paths that exists from s to the inputs of the circuit. In so doing, the algorithm may cause more switching/glitching

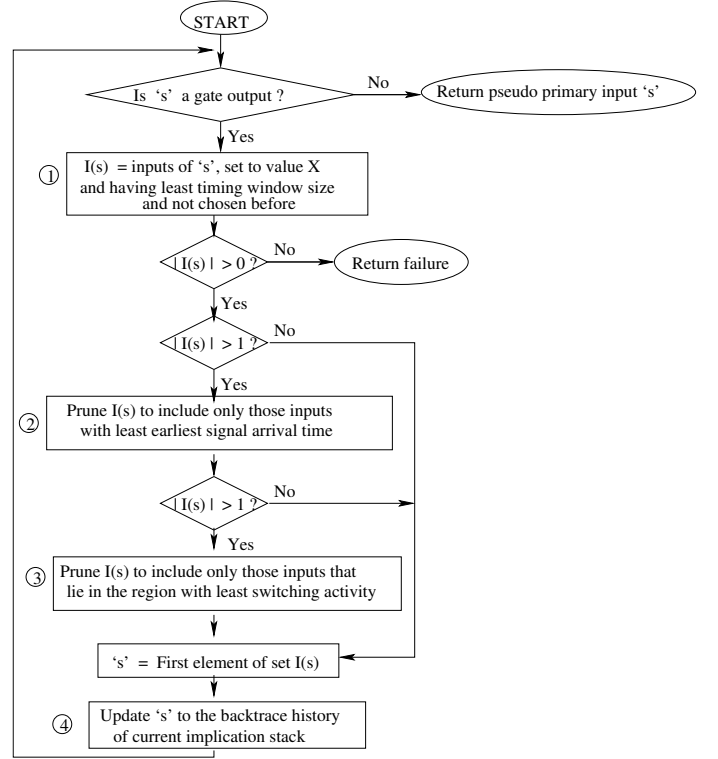
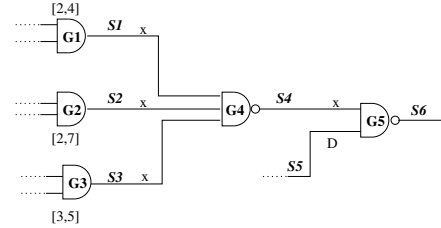


Fig. 5. Power-Aware heuristic for backtrace from fault location s to a pseudo primary input



LEGEND

- x – Unassigned value
- $[a,b]$ – Earliest signal arrival time is 'a', Latest signal arrival time is 'b'

Fig. 6. An example scenario for pattern generation

activity than required to achieve its function. Techniques have been proposed in [19] by using cost functions based on controllability and observability to minimize switching activity. In this work, we extend the Backtrace(s) and Objective() procedures of PODEM to be aware of the hazards and the regional power constraints. Figure 5 shows such an extension to the Backtrace algorithm. When s is an output of a gate G , step (1) considers the inputs to gate G which are currently set to X (unknown), as candidates to trace back to an input pin. The timing windows of these candidate pins are considered and the candidate with minimum sized timing window is selected for backtrace. Note that this greedy heuristic will minimize the switching activity at gate G , by reducing the possibility of hazard at gate G . If there are several candidates that contend for selection in step (1), the tie is broken in step (2) by selecting the gate input for which the arrival time is the least. This is illustrated in Figure 6, where the objective is to backtrace from the output of $G4$. The candidates for backtracing from $S4$ are $S1$, $S2$, and $S3$, all of which are currently set to X . The timing windows for these signals are $[2,4]$ and

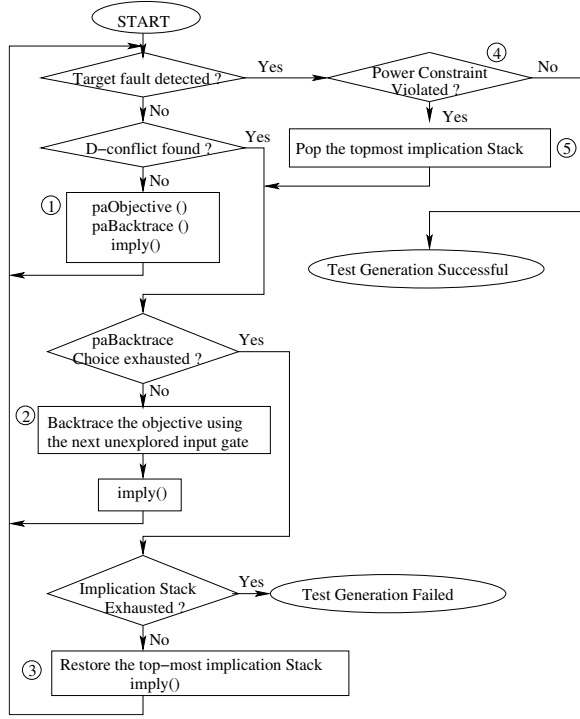


Fig. 7. Proposed Scheme for Pattern Generation

[3,5]. The timing window sizes are 2, 5, and 2, respectively. To break the tie between S1 and S3, we select S1 whose earliest arrival time ($t=2$) is smaller. Note that, in general, step (2) may not be able to resolve the tie; step (3) is used as a final tie-breaking rule by selecting a signal whose region has the least switching activity. For example, in Figure 6, if the timing window for S1 is also [3,5], then we will consider the switching activities in the regions to which gates G1 and G3 belong and break the tie. Finally step (3) may use randomization if it cannot use the regional switching activity to break the tie.

Rules corresponding to steps (1), (2), and (3) of the power-aware backtrace algorithm $paBacktrace(s)$ of Figure 5 are chosen to be applied in that order because of the relative contribution of timing window size, earliest arrival time, and regional switching activity towards the total impact on dynamic power drop, considering hazards.

We use a procedure called $paObjective()$ which uses similar power-aware decision making to derive an objective. When there are several candidate gates in the D-frontier [17], the gates are evaluated on their timing window size and $paObjective()$ selects the candidate gate having the least timing window size. If the above heuristic is unable to break ties, we attempt to break the tie using the switching activities in the regions to which the candidate gates belong. Finally, randomization is used to resolve the remaining ties.

Figure 7 shows the salient details of the complete power-aware ATPG algorithm. The heart of the algorithm is Step (1), which uses power-aware $paObjective(s)$ and $paBacktrace()$ procedures described earlier for selecting an objective and setting the inputs to achieve the objective. When the target fault is detected by the current pattern, the global and regional power constraints are validated before declaring success. If either global or regional power constraints has been violated, steps (2), (3) and (5) of the algorithm attempts to regenerate a different power-aware objective, uses a power-aware backtrace along a different path, and returns to check if the fault has been detected. Failure is declared if there is no path from the fault location to a primary input that has only X values and the error

TABLE II
INSTABILITY WINDOW PROPAGATION FOR 'AND' GATE

	0	1	R	F	0*	1*	R*	F*
			[c,c]	[c,d]	[c,d]	[c,d]	[c,d]	[c,d]
0	0	0	0	0	0	0	0	0
1	0	1	R	F	0*	1*	R*	F*
			[c,c]	[c,c]	[c,d]	[c,d]	[c,d]	[c,d]
R	0	R	R	0*	0*	R*	R*	0*
		[a,a]	[c,c]	[a,c]	[c,d]	[a,d]	[c,d]	[a,d]
F	0	F	0	F	0*	F*	0*	F*
		[a,a]		[a,a]	[c,b]	[a,b]	[c,b]	[a,b]
0*	0	0*	0*	0*	0*	0*	0*	0*
		[a,b]	[c,b]	[a,b]	[c,b]	[a,b]	[c,b]	[a,b]
1*	0	1*	R*	F*	0*	1*	R*	F*
		[a,b]	[c,d]	[a,d]	[c,d]	[a,d]	[c,d]	[a,d]
R*	0	R*	R*	0*	0*	R*	R*	0*
		[a,b]	[c,d]	[a,d]	[c,d]	[a,d]	[c,d]	[a,d]
F*	0	F*	0*	F*	0*	F*	0*	F*
		[a,b]	[c,b]	[a,b]	[c,b]	[a,b]	[c,b]	[a,b]

(D or D') has not propagated to an output. Note that the algorithm may fail because of two reasons : the fault is not detectable, or the fault is not detectable using a power-constrained pattern. Step (4) ensures that the fault is deemed to be detected only if the generated pattern does not violate power constraint.

We also implemented a dynamic compaction heuristic, which is an extension of the algorithm proposed in [11]. The modification is, while selecting a new target fault for the pattern, we select a gate that lies in the least active region. The pattern is then extended detecting the new fault iff global and regional power constraints are not violated.

3. Power-profiling and pattern optimization flow

1. Hazard-Aware Power-Profiling

Power-profiling of patterns is performed with due consideration to timing information so as to account for hazards. A window-based model was used to arrive at an optimistic estimate on the number of hazards in the circuit. Multiple hazards at a signal are abstracted using a window of earliest and latest signal instability times, called signal instability window. The interval denoted by $[a,b]$ in Figure 2 shows the signal instability window, considering static and dynamic hazards with unit delay model. The timing window used during pattern generation denotes the earliest and the latest possible signal arrival times that are obtained from STA in a vector-less manner. On the other hand, signal instability window is calculated for each pattern by using a window-based approach to approximate the exact time and number of hazard transitions. A hazard is identified if the size of the signal instability window at a gate is more than the inertial delay of the gate. The calculation and the propagation of the signal instability window is specific to the type of gate. Table II shows the computation of signal instability window at the output of an AND gate with 2 inputs, assuming that $a < c < b < d$. We use 8-valued logic, as explained in [17], for representing static and dynamic hazard. Thus, table II shows the corresponding 8×8 entries. As exact computation of all hazard transitions amounts to an expensive timing simulation of the circuit for each pattern, hazard calculation is approximated by attributing 2 transitions for each static hazard and 3 transitions for each dynamic hazard. Thus, instability window-based hazard analysis estimates the optimistic (or lower-bound) number of transitions in the presence of hazards without expensive simulations.

Once the hazard-based switching activity calculation for a pattern is complete, the toggle count at all the gates within each region is added to the toggle count of the respective region location. The regional

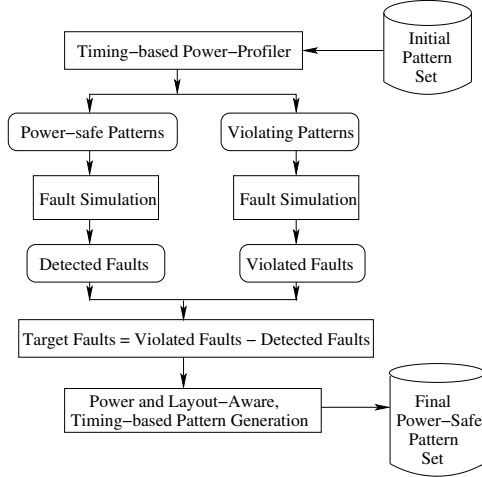


Fig. 8. Integrated Pattern Optimization Flow

toggle limit might be different for each region, due to irregular power grid topologies. The region toggle count for each region is compared against the respective regional toggle limit constraint, to identify unacceptably high localized switching activity. Then, the total toggle activity for all regions is compared against the global toggle limit constraint, to identify high overall switching activity. Patterns that fail one of these constraints are identified as violating patterns.

2. Integrated Pattern Optimization Flow

The proposed pattern optimization flow is illustrated in Figure 8. The initial pattern set, possibly from patterns generated by commercial ATPG tools, is profiled for power dissipation considering hazards. The pattern set is partitioned into two sets, viz. power-safe pattern set (patterns that satisfy both global and regional switching activity constraint), and power violating pattern set. Fault simulation is performed on each of the subsets. Pattern generation is targeted using the scheme proposed in Section 2.2, only for the violated faults that are not detected by the power-safe patterns. Thus, the flow aims to regenerate minimal number of patterns to obtain an optimized pattern set that satisfies power constraints.

4. Experiments and Results

The proposed algorithms for pattern generation, power-profiling, fault-grading and the optimization flow were implemented in Perl. Experiments were conducted on ISCAS 85 and 89 benchmarks with transition delay fault model. Run-times reported are for Linux AMD64 2.8GHz machine with 16 GB memory. Though the framework allows the use of actual timing information in lumped gate delay model, experiments were done with unit delay model for this paper. A single voltage and clock domain is assumed for all the experiments. Region size of each of the benchmark circuit was set to a square $N \times N$ region, and the value of N was chosen for each circuit in order to accommodate approximately 15 - 25 gates within each region. For regional toggle limit, a uniform value of approximately twice the maximum number of gates in a region was used, for all region locations. An increasing number of hazards have been observed and this could also be observed in the following experiments, where the peak number of toggles is sometimes much more than the gate count. As PODEM algorithm does implicit enumeration, abort limit of 100 failed re-tries was used to abort test generation for a target fault. A commercial ATPG tool was used to generate the initial pattern set.

The results are shown in Table III. The columns 1 to 4 denote the benchmark circuit, the initial pattern count generated by the

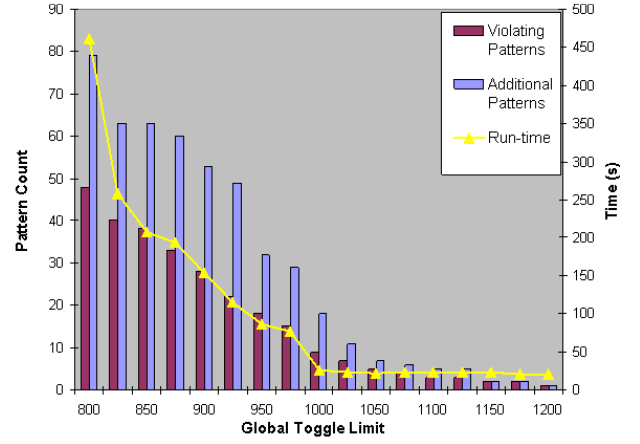


Fig. 9. Effect of Global Power Constraints for c1908

commercial ATPG tool, the peak capture toggles of the initial pattern set, and the maximum allowable global toggle limit, respectively. For all the circuits, the maximum allowable global toggle limit was set to 90% of the peak capture toggles. The number of patterns from initial pattern set violating the power constraints are given in column 5. The number of faults detected by the violating patterns that are not detected by the power-safe patterns are given in column 6. The loss in coverage if the violating patterns are not applied on the tester is also shown. Column 7 gives the number of additional power-safe patterns generated by the proposed power and layout-aware, timing-based ATPG tool to detect the violating faults. The increase in overall pattern count is also shown. Column 8 denotes the faults aborted by the proposed tool, and effective loss in coverage due to the aborted faults. Column 9 gives the new reduced peak switching activity, of the optimized power-safe pattern set. Columns 10 gives the total time taken for power-profiling, fault-grading and pattern generation. The peak switching activity in columns 3 and 9 were estimated using the instability window based power profiling.

To summarize, the proposed technique results in consistent decrease in the peak capture toggles of the optimized pattern set. Further, it may be noted that there is a small increase in pattern count. For the worst-case, the overall pattern count increased by 7.7% for s38417, while for the best case, the overall pattern count reduced by 2.3% for c1355. Further, the aborted faults contributed to negligible loss in coverage. Though the run-times are not very high, implementing in C/C++ would result in much faster run-times.

Figure 9 shows the effect of varying global power constraints for circuit c1908. It could be seen that as the global toggle limit is relaxed from 800 to 1200, the number of violating patterns, the number of additional patterns decrease, and the run-times significantly reduce.

To explore the effectiveness of region-specific local switching activity constraint, that exploits irregular power grid topologies, we did the following experiment on c1908 circuit. Consider a case, where the circuit has non-uniform functional activity as shown in figure 3. When power grid is over-designed for the worst case, uniform value of 45 for regional toggles would suffice, but this approach results in die area overhead. On the other hand, for optimal power grid uniform regional toggle limit of 35 would be correct, but it would unnecessarily over-constraint the patterns. Having a region-specific local toggle limit optimally constraints the pattern generation process. Figure 10 shows results upon having uniform regional toggle constraints of 45, 35 and allowing region-specific values varying

TABLE III
RESULTS FOR ISCAS BENCHMARK CIRCUITS

Circuit	Original Pattern Count	Original Peak Capture Toggles	Global Toggle Limit	Violating Patterns	Violating Faults (Cov. %)	Additional Patterns (% Incr.)	Aborted Faults (Cov. %)	New Peak Capture Toggles	Total CPU Time(s)
c1355	216	621	558	18	26 (1.2%)	13 (-2.3%)	4 (0.2%)	556	49
c1908	85	1215	1093	3	6 (0.2%)	5 (2.4%)	0 (0.0%)	1089	21
c2670	84	1658	1492	8	66 (1.6%)	14 (7.1%)	11 (0.3%)	1466	116
c3540	180	2178	1960	5	34 (0.6%)	13 (4.4%)	7 (0.1%)	1957	156
c5315	116	2616	2354	9	44 (0.5%)	12 (2.6%)	2 (0.0%)	2350	138
c7552	129	4099	3689	5	28 (0.2%)	13 (6.2%)	0 (0.0%)	3676	181
s1423	45	685	616	1	5 (0.2%)	3 (4.4%)	0 (0.0%)	612	10
s5378	150	2079	1871	6	83 (1.0%)	14 (5.3%)	0 (0.0%)	1855	128
s9234	188	4569	4112	5	179 (1.1%)	18 (6.9%)	0 (0.0%)	4105	541
s13207	316	6556	5900	2	124 (0.6%)	15 (4.1%)	0 (0.0%)	5752	647
s15850	163	9989	8990	3	69 (0.3%)	14 (6.7%)	1 (0.0%)	8705	474
s38417	156	18882	16993	6	317 (0.5%)	18 (7.7%)	3 (0.0%)	16827	2659
s38584	266	15987	14388	4	204 (0.3%)	13 (3.4%)	4 (0.0%)	13948	1934

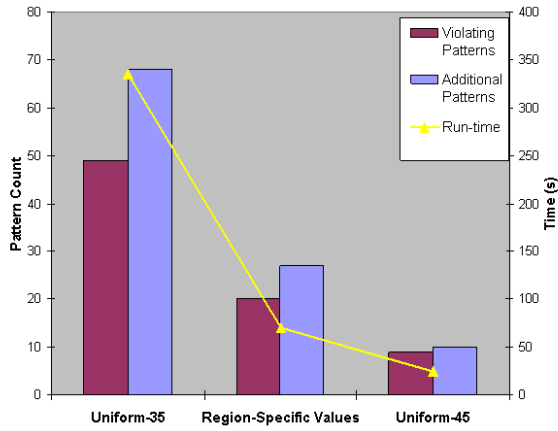


Fig. 10. Effect of Regional Power Constraints for c1908

from 35 to 45 based on the region. It may be noted from the figure that uniform constraint of 45 has the least additional pattern count and minimal run-time. This could be used when power grid is over-designed. For optimal power grid, over-constraining the ATPG flow with uniform value of 35 results in excessive pattern count and run-time. Region-specific values provides the best results with minimal additional patterns and reduced run-time.

5. Conclusion and Future Work

Most of the commercial ATPG tools and existing techniques proposed for low power pattern generation do not consider timing information. In this paper, we observe an increasing contribution of hazards to the total dynamic power dissipation and the lack of correlation between patterns that dissipate high peak power with zero and unit delay models. Techniques are proposed in this paper for timing-based power profiling of the patterns, considering the effects of hazards. We proposed power and layout-aware pattern generation to minimize global and regional power dissipation. The proposed technique also comprehends the effect of irregular power grid topology by allowing non-uniform limits on regional switching activity. Further, an integrated flow is proposed for obtaining an optimized power-safe pattern set, with minimal additional patterns.

Future work includes optimizing the run-time by implementing the proposed flow in C++. Further, detailed power rail analysis is planned to evaluate the effectiveness of the proposed technique in minimizing

the IR-drop on the power grid.

References

- [1] J. Saxena, *et al.*, "A Case Study of IR-Drop in Structured At-Speed Testing," in *Proc. Intl. Test Conf.*, 2003, pp. 1098–1104.
- [2] K. M. Butler, *et al.*, "Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques," in *Proc. Intl. Test Conf.*, 2004, pp. 355–364.
- [3] C. Shi and R. Kapur, "How Power-Aware Test Improves Reliability and Yield," *EE Times*, Sep 2004.
- [4] K. Shakeri and J. D. Meindl, "Compact Physical IR-Drop Models for Chip/Package Co-Design of Gigascale Integration," *IEEE Trans. on Electron Devices*, pp. 1087–1096, June 2005.
- [5] A. Kokrady and C. P. Ravikumar, "Fast, Layout-Aware Validation of Test-Vectors for Nanometer-Related Timing Failures," in *Proc. Intl. Conf. on VLSI Design*, 2004, pp. 597–602.
- [6] P. Girard, "Survey of Low-Power Testing of VLSI Circuits," *IEEE Design & Test of Computers*, pp. 82–92, May/June 2002.
- [7] 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 of Integrated Circuits and Systems*, pp. 1142–1153, July 2004.
- [8] S. Bhunia, *et al.*, "Low Power Scan Design Using First Level Supply Gating," *IEEE Trans. on VLSI Systems*, pp. 384–395, Mar 2005.
- [9] T. Yoshida and M. Watari, "MD-SCAN Method for Low Power Scan Testing," in *Proc. Asian Test Symp.*, 2002, pp. 80–85.
- [10] W. Li, S. M. Reddy, and I. Pomeranz, "On Reducing Peak Current and Power during Test," in *Proc. IEEE Comp. Society Annual Symp. on VLSI*, 2005, pp. 156–161.
- [11] X. Wen, *et al.*, "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing," in *Proc. VLSI Test Symp.*, 2006, pp. 58–63.
- [12] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "Effects of delay Models in Peak Power Estimation of VLSI Sequential Circuits," in *Proc. Intl. Conf. Computer-Aided Design*, 1997, pp. 45–51.
- [13] V. D. Agrawal, "Low-Power Design by Hazard Filtering," in *Proc. Intl. Conf. on VLSI Design*, 1997, pp. 193–197.
- [14] S. Uppalapati, M. L. Bushnell, and V. D. Agrawal, "Glitch-Free Design of Low Power ASICs Using Customized Resistive Feedthrough Cells," in *Proc. VLSI Design And Test Symp.*, 2005, pp. 41–48.
- [15] B. Kruseman, *et al.*, "On Hazard-free Patterns for Fine-delay Fault Testing," in *Proc. IEEE Intl. Test Conf.*, 2004, pp. 213–222.
- [16] T. Meyyappan, V. Visvanathan, and S. K. Nandy, "Robust Power Delivery for Sub-100nm Integrated Circuits," *Embedded Tutorial, VLSI Design And Test Symp.*, 2006.
- [17] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Computer Science Press, 1990.
- [18] X. Liu, *et al.*, "Efficient Transition Fault ATPG Algorithms Based on Stuck-At Test Vectors," *Journal of Electronic Testing: Theory and Applications*, pp. 437–445, 2003.
- [19] S. Wang and S. K. Gupta, "ATPG for Heat Dissipation Minimization During Test Application," *IEEE Trans. on Computers*, pp. 256–262, 1998.