Capture Power Reduction Using Clock Gating Aware Test Generation

Krishna Chakravadhanula, Vivek Chickermane, Brion Keller, Patrick Gallagher and Prashant Narang
{ckrishna, vivekc, kellerbl, patrickg, pnarang}@cadence.com
Front-End Design Group, Cadence Design Systems, USA

Abstract

Scan-based manufacturing test of low power designs often exceeds the very tight functional constraints on average and instantaneous logic switching. The logic activity during the shift and launch-capture of test pattern data may lead to excessive power consumption and voltage droop. This paper focuses on the management of instantaneous power during the capture phase. By taking advantage of the existing clock gating circuitry and selectively holding the value of some scan flip-flops, switching activity during the capture cycles of a test can be reduced. The effectiveness of this technique is demonstrated on several industrial designs that show up to 30% (55%) reduction in instantaneous (average) capture switching.

1. Introduction

Test mode power can exceed the functional power specifications for several reasons cited below. This can lead to excessive IR, inductive voltage drop, and thermal problems during manufacturing test and result in lower yield.

1) Low Power (LP) enabling circuitry such as power-shut off (PSO), isolation gates, etc., may be disabled during manufacturing test.
2) In scan designs the automatic test pattern generation (ATPG) tool may use random fill techniques to set the "don't care" bits leading to about 50% activity level in the flops. This is significantly higher than the 7-10% functional activity for many LP designs
3) All the clocks must be active in each scan cycle and since the clocks may have simultaneous edges there can be a significant increase in instantaneous power.
4) ATPG compaction algorithms maximize the faults detected in each vector. This will typically increase logic activity.
5) Delay tests that utilize "Launch off capture" transition approaches can cause voltage drop due to inductive effects as two high speed clock pulses follow a quiescent period.
6) A larger percentage of the clock gating logic may be enabled than functionally acceptable. This will increase both the activity level of the logic driven by the flip-flops and the power consumed in the clock network which can be up to half the dynamic power.

University and EDA industry researchers have developed many innovative power-aware DFT and ATPG methods [1-8]. Clock gating DFT techniques for LP Test in general is well described in [6]. A technique for Multi-Supply Multi-Voltage (MSMV) and PSO designs described by industry standard power specifications addresses issue 1 [9-11]. The use of intelligent X-fill techniques to reduce scan and capture switching has been addressed by some methods [7,17,18]. Using these techniques we can reduce the average activity during scan shift but this may result in a "saw tooth" distribution of activity level over time [16]. Immediately after the capture cycle, the flops will have loaded capture values generated by the functional logic. These will be dependent upon the scan load values and the functional logic. However, they will normally be more random in 0/1 distribution than the scan load pattern. At the start of the shift operation starts we will see an activity level around 50%. As the "capture values" are unloaded and the repeat-fill load values fill the scan chains we will see the activity level drop until the next capture cycle. So the "pattern fill" strategies alone will not solve either the peak IR drop problem or the average IR drop if we measure this across a window of tens of clock cycles or less.

This work addresses the domain of capture power reduction within an ATPG tool without relying on X-fill techniques that lead to the problem described above. The paper is organized as follows. The next section describes prior work in capture power reductions. Section 3 analyzes capture activity for many industrial designs which motivates the Clock Gating ATPG methodology described in Section 4. Experimental results are provided in Section 5 and more analyses based on some additional experiments are provided in Section 6. We then conclude with a summary of the work and provide some future directions for research.

2. Previous Work

Previous work to reduce capture power can be broadly classified under two categories. One approach has been to perform X-fill of the don't-care bits in test patterns so that flop transitions during the capture cycle are reduced. This approach is limited by the number of care bits that can be added to the test patterns. Designs supporting test data compression are becoming increasingly common at

978-1-4244-4867-8/09/$25.00 ©2009 IEEE

smaller technologies, and compression techniques cannot handle many care bits in the same scan shift cycle.

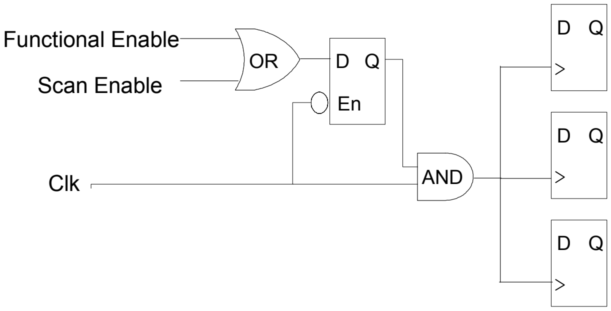


Figure 1. Example of Clock Gating

More recently an alternate approach has been proposed that uses the existing clock gating logic in the design to hold the clock controlling the flops that are not required to switch within a given test pattern [14-16]. Figure 1 shows a typical clock gate inserted by LP synthesis tools. During scan shift the scan enable signal overrides the clock gate and allows all flops to clock concurrently. During the capture cycle the scan enable is inactive and the clock gate is controlled from its functional enable. If the functional enable can be selectively disabled in concordance with the ATPG algorithm then capture power can be reduced without impact to the ability to capture fault effects.

The proposed solution to these problems is to limit the clock activity in the capture cycles by imposing ATPG constraints on the enable signals of the clock gates. This helps in several ways:
1) Up to 50% or more of the dynamic power distribution is in the clock nets so there is a great reduction in clock tree power.
2) Fewer flops will switch during capture, further reducing the logical switching and IR drop.
3) The flops which do not update will keep the "low activity" scan shift pattern that was loaded. So when the next shift starts we will have an initial pattern which is composed of a combination of random capture values and the repeat-fill pattern. This will reduce the initial activity during shift and hence reduce the "saw tooth" distribution of scan shift activity over time.
4) This approach is more compatible with test data compression since only a few care bits may be sufficient to hold the clock at an internal node that controls a large number of flops
5) Reduces the impact on test pattern compaction algorithms that try to subsume several patterns by taking advantage of don't-care bit positions.
6) Does not rely on limiting the ATPG to use only a subset (or single) clock to capture between each shift since this gives unacceptable increases in test time and data sizes.

In addition to [16] some recent papers [12-13] have also considered the use of clock gating for capture power reduction. One of the key differences in this paper relative to [12] is that we implement and describe a more comprehensive and empirical clock gating analysis. We apply methods similar to those used by LP synthesis tools to consider clock gating hierarchies, fine vs. coarse grained gating, and the type of clock gating library cell inserted during LP synthesis. This allows us to address capture power issues in several industrial designs after their clock networks are physically implemented. Only the clock gates on the clock(s) being pulsed in the test pattern are considered. After sufficient clock gates have been turned off to bring the switching below a user specified threshold, the remaining don't-cares in the test pattern are filled using intelligent X-fill techniques to ensure that the load and unload shift power is also minimized. The key difference with [13] is that we have enhanced the ATPG algorithm itself rather than do a post-ATPG analysis which may lead to inefficiencies.

3. Analysis of Capture Activity

During the capture cycle of test patterns, the switching activity at a flip-flop can be mapped to one of the four scenarios shown in Table 1. These four scenarios exist once a pattern is generated by the test generator and before the don't-cares (X's) in the pattern are filled with a known value (0/1). Only flops required for fault detection will contain 'care-bits' (either 0 or 1), and the remaining flops are left as don't-cares by the test generator. Don't-cares in the scan load portion of the test pattern are set to a known value based on various 'X-fill' techniques. After filling the scan load don't-cares the test pattern are simulated and this causes the don't-cares in the scan unload to be at a known value. In this analysis we do not include X's in the scan unload caused by the capture of X-sources in the design.

In Table 1, Scenario A is where the flop is a don't-care in the scan load, but captures a care-bit once the functional clock is applied after scan shift. This care-bit is required for detection of a fault effect. Scenario B is where the flop has a care-bit for excitation of the fault(s), and a care-bit for fault detection. In scenario C the value captured in the flop is not required for fault detection. In scenario D, the flop is not required for stimulus or observation of any of the faults targeted by that pattern. Figure 2 illustrates the four scenarios in Table 1 using an example that has 4 flops in a scan chain. Based on the classification above different approaches are possible to reduce capture switching activity.

1) Scenario A is computationally the most simple to address by ensuring that the don't-care in the scan load is filled with the same care-bit value present for that flop in

the scan unload. Since the flop will now be loaded with the same value it is supposed to capture after the functional capture cycle, there will be no switching happening at the flop output.
2) Scenario B provides no scope for experimentation. If the care-bit loaded is different from the care-bit value after the capture cycle, there will be a transition at the flop output.

Table 1. Scenarios at a flop before and after the Capture cycle

Scenarios	Flop(i) value after Scan Load	Flop(i) value in Scan Unload
A	X	Care bit
B	Care bit	Care bit
C	Care bit	X
D	X	X

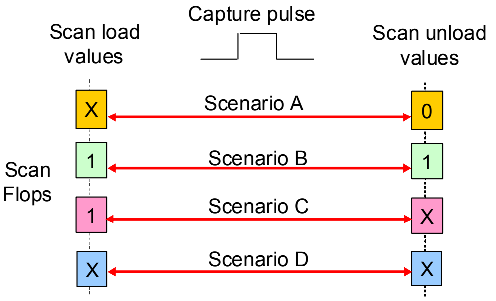
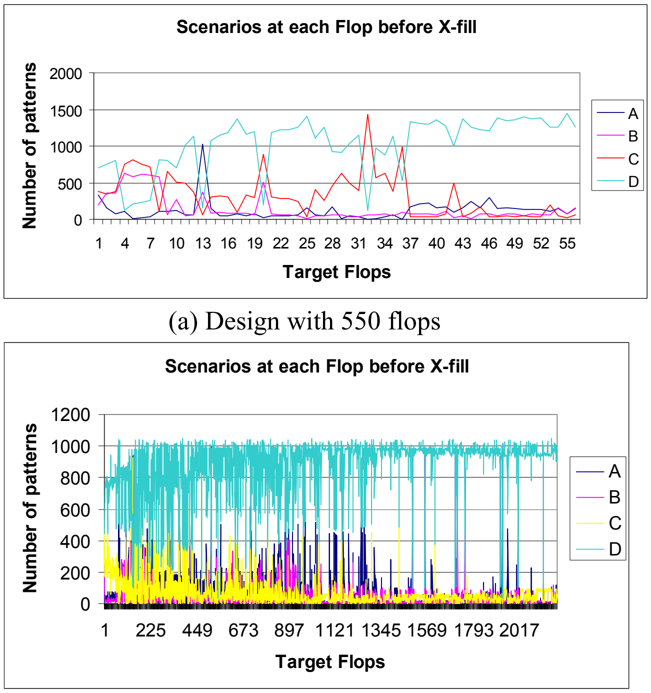


Figure 2. Example Capture scenarios shown for 4 flops

3) Scenario C can only be addressed by sophisticated X-fill techniques as the flop is already scan loaded with a care-bit. One approach could be to perform additional test generation to ensure the flop captures the same value it is loaded with. This may lead to increased care-bit density in the loaded pattern and additional switching during scan load. A simpler approach, where possible, would be to gate the clock to this flop during the capture cycle so that it does not capture a new value.
4) Scenario D provides scope for different heuristics to be applied and can end up as the most computationally expensive when looking at filling the don't-cares to reduce capture power. One approach could be to apply techniques that perform X-fill to reduce capture power. An alternative would be to gate off the clock to such flops as they are not required for fault detection.

Experiments were performed on some industrial designs to understand how often each scenario occurs in a typical test pattern set. X-fill techniques that reduce capture power can operate only on a subset of the total flops, as they would otherwise add too many care-bits to the test patterns. For the purpose of these experiments, the flops were sorted based on the impact a transition at their output could have on the switching activity of the design. The top 10% of the flops that had the most impact on the switching activity were selected. Figure 3 shows the results of these experiments for two designs, one having 550 flops, and a bigger design having 22K flops.



(a) Design with 550 flops

(b) Design with 22K flops

Figure 3. Frequency of occurrence of each flop scenario

Both these designs showed similar results. Scenario D was the most common, followed by scenario C. Table 2 summarizes the key learnings from these experiments. *What is important to understand here is that out of the two most common scenarios C and D, simple and efficient X-fill techniques for reducing capture power mainly address scenario D, whereas ATPG-based techniques that gate off the clocks can be effective for both scenarios C and D.*

Table 2. Possible techniques for each scenario at a flop

Scenarios	**Frequency of occurrence**	**Techniques to Reduce Capture Switching**
A	Low	X-fill for capture power
B	Low	N/A
C	Medium	Clock Gating aware ATPG
D	Very High	X-fill for capture power OR Clock Gating aware ATPG

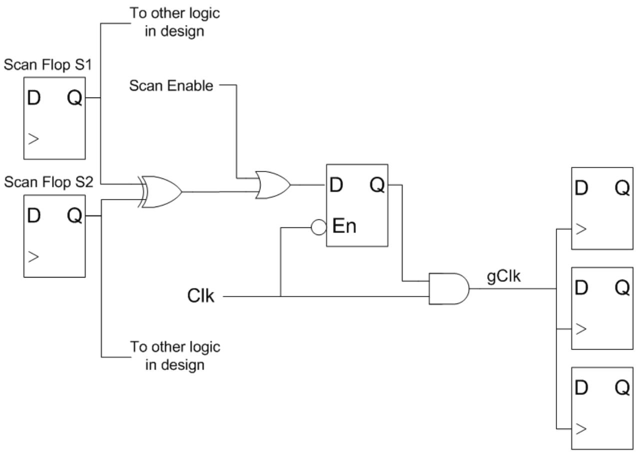
4. Clock Gating ATPG Methodology

This section describes the proposed approach called Clock Gating-aware ATPG (CG-ATPG) to reduce capture power by using clock gates present in the design. To turn off clock gates to reduce capture switching, we need to first locate the clock gates in the design. Once the clock gates in the design are identified, test generation is done to obtain a 'default' set of care bits required to gate the clock off at each clock gate. While there could be multiple ways of turning off a clock gate, currently only one solution (called 'default'), is stored. For each test pattern generated by the test generator, where possible, we then merge the default care bits of each clock gate. If the care bits for a clock gate conflict with the care bits in the test pattern, the next clock gate is tried, and so on. Note that only the clock gates on the clock(s) being pulsed in the test pattern are considered. After enough clock gates have been turned off to bring the switching below a user specified threshold, the remaining don't-cares in the test pattern are filled using techniques like *repeat-fill* for scan shift power reduction.

Figure 4 shows an example [19] of how the 'default' care bits to gate the clock off are merged with the test cubes. Other than the Scan Enable being logic-0 during the capture cycle, the default care bits 'S1=0, S2=0' are sufficient to gate the clock off at the clock gate in the figure. These clock gate care bits are compared against the care bits in the three test cubes T1, T2, and T3 to see if the clock 'Clk' can be gated off at this clock gate. Figure 4(c) shows that the 'default' care bits can be merged successfully into test cubes T1 and T3, but not for T2. The scan flop S2 was a don't-care bit in T1, and after merging has a care-bit '0'. Similarly for flop S1 in T3. For test cube T2, the value chosen for X by *repeat-fill* (or similar algorithms) will determine if the clock is gated off at that clock gate.

It is possible that the initial test cube – before any clock gate care bits are merged – may already have a high number of care bits in it. If dynamic compaction during ATPG merges the care bits for a large number of target faults, it can increase the care bits significantly. We may then not be able to exploit a sufficient number of clock gates due to conflicts with existing care bits. This can be addressed via the alternatives mentioned below.

One approach could be to delete those patterns having high capture switching activity since sufficient number of clock gates could not be turned off. To target the faults detected by the deleted patterns, new patterns can be generated using a lower compaction effort during test generation. While this process shown in Figure 5 may have to be done iteratively to achieve the desired fault coverage, it helps in ensuring the pattern count does not increase significantly.



(a) Gating logic controlled by two scan flops. Default values are S1=0, S2=0.

Test Cubes	Scan flops in design					
	S0	S1	S2	S3	S4	S5
T1	X	0	X	1	0	X
T2	0	1	X	X	X	0
T3	X	X	0	1	0	X

(b) Test cubes containing care bits from ATPG

Test Cubes	Scan flops in design						Merge "default values"?
	S0	**S1**	**S2**	S3	S4	S5	
T1	X	**0**	**0**	1	0	X	Yes
T2	0	**1**	**X**	X	X	0	Conflict
T3	X	**0**	**0**	1	0	X	Yes

(c) Test cubes after adding "default values"

Figure 4. Example of clock gate 'default' care bits merged into ATPG test cubes

Another approach would be to merge in the clock gate care bits during dynamic compaction. Before the care bits for the next target fault are merged, analysis is done to see if too many clock gates are enabled. If so, some clock gates can be turned off (by merging their care bits) before the care bits for the next target fault are merged into the existing test pattern. While this can increase the runtime for test generation, the advantage lies in being able to generate patterns that lie within the switching limits desired, without going through an iterative process.

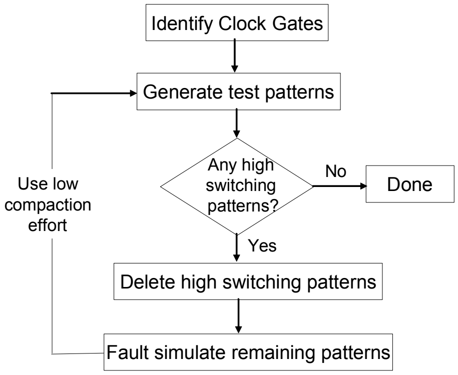

Identify Clock Gates
Generate test patterns
Any high switching patterns?
No
Done
Yes
Use low compaction effort
Delete high switching patterns
Fault simulate remaining patterns

Figure 5. Generating patterns within switching limits

It is possible that the 'default' care bits identified for a clock gate may conflict with a large number of test patterns. An alternative is to dynamically generate the clock gate care bits for each test pattern by invoking the test generator separately so that, if possible, clock gate care bits compatible with the current test pattern might be derived. While this approach can also increase the runtime, it allows for potentially better results to be achieved. For the example in Figure 4, the solution 'S1=1, S2=1' would be compatible with T2, resulting in the clock gate being mergeable in all three test cubes.

4.1. Locating Clock Gates in the Design

Synthesis support for clock gate insertion has become very sophisticated and current designs may contain within them coarse and/or fine grain clock gating as shown in Figure 6. While a coarse grain clock gate may control thousands of flops, a fine grain clock gate controls only tens of flops. The granularity of clock gates is important since by gating the clock off at a coarse grain clock gate we may be able to prevent thousands of flops from switching, and it may be possible to perform this gating using only a few care bits.

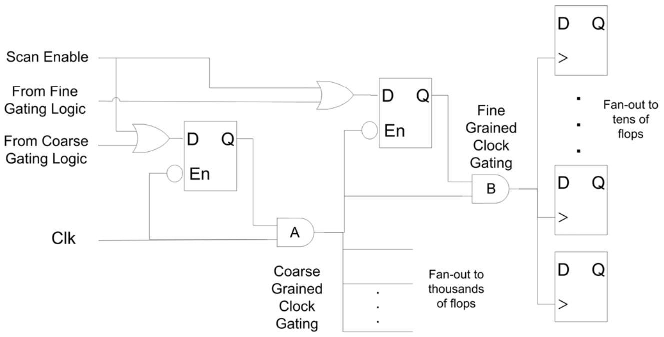


Figure 6. Fine and coarse-grain clock gating

The process of identifying clock gates involves tracing back from the clock pin on each flop in the design. As each clock gate is encountered during such structural tracing, information is stored about the number of flops controlled by that clock gate. For designs having both coarse and fine-grain clock gating, there could be multiple clock gates encountered before the root clock is reached. If so, a clock gate could be controlling not only flops but other clock gates as well. Such information about the clock gate hierarchy is stored as well. This also improves performance since the next time a clock gate is seen, there is no need to travel up the hierarchy since that has already been done the first time around. Note that clock gates that are connected to a 'test mode enable' signal – that stays at a constant value during test, unlike 'scan enable' as shown in Figure 1 – are ignored by this approach since they can never gate the clock off during the capture cycles.

Among the information tagged with each clock gate and root clock in the design is the number of flops controlled directly and via child clock gates, the child clock gates controlled and the care bits to gate off the clock at the clock gate. Information is also stored as to which clock it gates, since we do not want the test generator to spend time and add care bits for turning off a clock gate unless its clock is being pulsed in that test pattern.

Once the clock gates have been identified, they are sorted in descending order based on the total number of flops controlled by each clock gate. This is to ensure that the test generator first tries to gate the clock at a coarse-grain clock gate. If successful, all the fine-grained child clock gates under this coarse-grain clock gate can be skipped for that test pattern.

It is important to know what percentage of the flops on a clock tree are gated. Let us take for example a clock that controls 100K flops, out of which 70K can be controlled via clock gates. Even if all clock gates can be turned off to gate the clock, 30K flops would still be able to switch during the capture cycle. Assuming the worst case where every flop transitions during the capture cycle, we could see 30% capture switching activity. If only 50K flops receive gated clocks, then even after the best efforts on the part of CG-ATPG the user could still see capture switching of up to 50%. If a clock tree has a large percentage of flops not driven by a clock gate, it will limit the effectiveness of clock gating based techniques for capture power reduction. In some cases, it may be necessary to insert additional clock gates from a DFT perspective to ensure that capture switching activity does not exceed the power budget of the design.

5. Experimental Results

This section describes results from experiments that were done on some industrial designs. Table 3 shows the characteristics of these designs and some statistics on the clock gating present in these designs. Design A has four clock domains and has 1340 clock gates that allow both coarse and fine-grain clock gating. Design B has ten clock domains but only five of these domains contain the combined total of 4382 clock gates. All these are fine-grain clock gates, each requiring a relatively higher number of care bits to gate the clock off as compared to the other designs. Design C has one clock domain having 3136 clock gates. Only a few of the fine-grain clock gates are controlled by a coarser-grain clock gate; most of the fine-grain clock gates are driven directly by the root clock.

Table 3. Designs used for Clock Gate analysis

Design	# flops	# clock gates	% of total flops gated	Average number of flops per clock gate	Average care bits per clock gate
A	170K	1340	27.8%	34.3	4.5
B	150K	4382	74.6%	25.5	12.77
C	46K	3136	82.7%	12.1	5.58

Patterns were then generated for these designs using both the proposed Clock Gating aware test generation (CG-ATPG) and conventional test generation using a commercial ATPG tool. The capture switching activity was calculated as the percentage of flops that toggle during the capture cycle. It is assumed the flop switching activity will closely estimate the gate or transistor level switching activity.

Figure 7 shows how switching during the capture cycle varies across the generated test patterns. Results are shown for patterns where the don't-cares are filled using *repeat-fill*. CG-ATPG achieves lower capture switching than conventional ATPG. A similar graph with an even larger decrease in switching activity is obtained for patterns with random-fill since the flops whose clocks are gated off continue to hold the values that were scan loaded. In conventional ATPG these flops are not guaranteed to hold state, and hence it is likely they would have higher switching in them for random-fill patterns as compared to repeat-fill.

Table 4 shows a comprehensive set of results for capture switching activity for the three designs. Patterns are generated for the stuck-at fault model using the CG-ATPG approach wherein some don't-cares are filled with the clock gate care bits and remaining don't-cares are filled with repeat-fill values. When using conventional ATPG, all the don't-cares are filled with repeat-fill values. The column 'Average Instantaneous Capture Switching' reports switching in the capture cycle when averaged across all the generated patterns. The column 'Maximum Instantaneous Capture Switching' reports the maximum switching in a capture cycle when seen across all the patterns. The last two columns of Table 4 show the percentage reduction in capture switching activity when CG-ATPG is used when compared to using conventional ATPG.

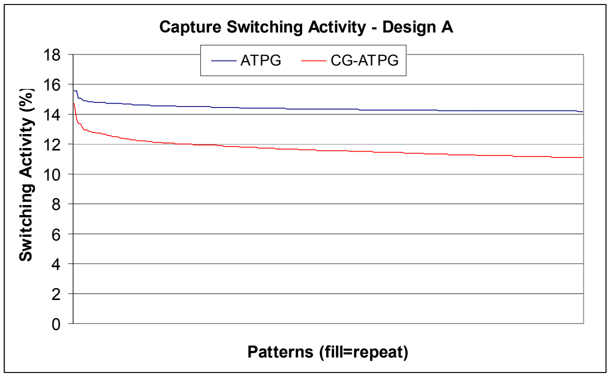


Figure 7. Capture Switching activity comparison when patterns are generated using **fill=repeat**

As can be seen from the table, all three designs show significant reduction in both maximum and average instantaneous capture switching activity. Initial experiments showed little or no reduction in maximum instantaneous switching activity. Upon analysis it was found that these peaks were primarily due to patterns that were detecting faults on asynchronous set/reset signals that control a large number of flops. These signals do not have any clock gates on their path that can be exploited by CG-ATPG; hence switching activity was the same for both the test generation approaches. This highlights the importance of building enough skew in the fanout cone of the reset signal so that not all its flops switch at the same time. An alternative would be to improve pattern generation such that multiple patterns are generated and only a subset of these flops is allowed to switch in each pattern.

To correctly determine the benefits of CG-ATPG in reducing peak capture switching, we excluded these faults from the ATPG runs that were done to generate the experimental results shown here. For fair comparison, these faults were excluded from the conventional ATPG runs as well. While designs A and C show good reduction in both average and peak capture switching, design B provides the best results.

To understand the impact of pattern compaction on the CG-ATPG approach, separate test generation runs were done using low and high compaction effort levels. Since higher compaction would increase the number of care bits in each pattern generated by the test generator, it increases the likelihood of conflicts when merging the clock gate care bits. While pattern compaction does not seem to have much impact on the average capture switching, column 6 in table 4 shows the maximum capture switching increasing for higher compaction effort. Hence the reduction in maximum capture switching (last column) is less for higher compaction effort.

As expected, higher reduction in switching activity was observed for patterns generated using random-fill. Note that in reality it is desired that the don't-care fill applied be somewhere in-between repeat-fill and random-fill. Using repeat-fill tends to under-stress the design from a power perspective, whereas random-fill can over-stress it. The ideal scenario would be to have the switching during test application be close to or a little higher than the functional switching activity; or the activity that can be tolerated by the power grid. Our experimental setup has the ability to intelligently vary the X-fill percentage to meet the user specified switching limits.

Table 4. Results for **CAPTURE switching** activity in **FULLSCAN** patterns generated with **fill=repeat**

Design	Pattern Compaction effort	Capture Switching activity (**Conventional ATPG**) (%)		Capture Switching activity (**CG-ATPG**) (%)		**Reduction** (%)	
		Average Instantaneous Capture Switching	Maximum Instantaneous Capture Switching	Average Instantaneous Capture Switching	Maximum Instantaneous Capture Switching	Average Instantaneous Capture Switching	Maximum Instantaneous Capture Switching
A	Low	10.69	15.58	8.15	14.74	23.8	5.4
	High	10.6	16.08	8.09	15.08	23.7	6.2
B	Low	3.55	32.92	1.55	22.25	56.3	32.4
	High	3.52	32.83	1.56	25.81	55.7	21.4
C	Low	19.53	27.21	12.3	21.99	37.0	19.2
	High	19.34	27.21	12.59	24.63	34.9	9.5

Table 5. Results for **SCAN SHIFT switching** activity in **FULLSCAN** patterns

Design	Pattern Compaction Effort	Scan Switching in Patterns with **fill=repeat**						Scan Switching in Patterns with **fill=random**					
		Conventional ATPG (%)			**CG-ATPG** (%)			**Conventional ATPG** (%)			**CG-ATPG** (%)		
		Scan In	Scan Out	Total Shift	Scan In	Scan Out	Total Shift	Scan In	Scan Out	Total Shift	Scan In	Scan Out	Total Shift
A	Low	2.16	4.76	6.92	2.13	4.22	6.35	23.73	22.31	46.04	23.72	22.43	46.15
	High	2.16	4.74	6.9	2.12	4.21	6.33	23.73	22.32	46.05	23.72	22.44	46.16
B	Low	2.23	3.4	5.63	2.08	3.69	5.77	24.97	13.84	38.81	24.87	23.25	48.12
	High	2.21	3.39	5.6	2.07	3.7	5.77	24.96	13.99	38.95	24.86	23.28	48.14
C	Low	2.15	6.1	8.25	2	5.24	7.24	24.34	20.6	44.94	24.2	22.96	47.16
	High	2.12	6.09	8.21	1.98	5.24	7.22	24.32	20.71	45.03	24.18	22.92	47.1

Table 5 shows the impact on scan shift switching activity when the proposed CG-ATPG method is used. The switching activity during scan is separated into the activity caused by loading the test pattern (Scan In) and the activity caused by unloading the circuit response (Scan Out). Since more care bits are added to the test pattern, the scan in switching activity is expected to increase. On the other hand, the care bits merged into the patterns are now preventing some flops from switching during the capture cycle. These flops will hold the values they were loaded with during scan in. If the scanned in values have low switching, this will help reduce the scan out switching as well. Patterns generated using repeat-fill exhibit this behavior well for designs A and C.

For patterns using random-fill, the scan out switching could actually increase due to the following phenomenon. If the functional switching activity of the circuit is less than the switching activity of random-fill (50%), then the flops prevented from switching during capture will continue to hold the random-fill data, while the remaining flops will capture from the functional logic. These flops holding the random-fill data can cause the overall scan out switching to be higher when compared to the conventional case. While this is exhibited slightly by designs A and C, it is very prominent in the case of design B, whose scan out switching greatly increases for random-fill patterns.

Table 6. ATPG statistics for **FULLSCAN** patterns generated with **fill=repeat**

Design	Pattern Compaction effort	Pattern Count Increase (%)	Drop in Fault Coverage	Increase in Runtime (%)
A	Low	0.2	0.02%	6.33
	High	-0.19	0.03%	0.75
B	Low	15.07	0.03%	50.94
	High	3.04	0.03%	13.75
C	Low	8.29	0.11%	-1.47
	High	1.62	0.14%	2.28

Table 6 shows the increase in pattern count, drop in fault coverage, and increase in runtime for CG-ATPG when compared to using conventional ATPG. For all the cases, except design B (with low compaction effort), the percentage increase in pattern count is less than 10%, while the increase in runtime is less than 15%. There is negligible drop in fault coverage due to the CG-ATPG algorithm. For design A (with high compaction effort) the number of patterns generated dropped marginally when using CG-ATPG. Similarly for design C (with low compaction effort), the runtime for CG-ATPG was slightly less than that for conventional ATPG.

Table 7 shows capture switching results for the two designs that had test data compression structures in them. The compression architecture has a combinational de-compressor on the scan-in side, and an XOR-based compactor on the scan-output side. Once again, CG-ATPG provides very good results for design B by significantly reducing both the average and maximum capture switching activity. The increase in pattern count and runtime is within the same range as seen for FULLSCAN, and the decrease in fault coverage is minimal.

Table 7. Results for **CAPTURE switching** activity in **COMPRESSION** patterns generated with **fill=repeat**

Design	Pattern Compaction effort	**Reduction** in Capture Switching (%)		ATPG statistics		
		Average Instantaneous Capture Switching	Maximum Instantaneous Capture Switching	Pattern Count Increase (%)	Drop in Fault Cov.	Increase in Runtime (%)
A	Low	25.6	5.4	3.2	0.03%	2.71
	High	25.6	2.7	-0.19	0.03%	-1.22
B	Low	50.3	34.5	47.56	0	38.03
	High	47.4	25.3	11.13	0.01%	9.7

6. Ordering the Clock Gates

For the results shown above, the clock gates identified in the design were sorted in descending order based on the total number of flops controlled by each clock gate. To understand if the switching activity is impacted by the sorting, a couple of other heuristics were tried as well.

a. Number of flops controlled by each clock gate
b. Number of flops controlled per clock gate care bit
c. Unsorted list of clock gates

The motivation behind considering (b) is that there could be two clock gates each controlling 100 flops, but while the first clock gate may require 10 care bits to gate the clock off, the second clock gate may require only 2 care bits. It would make sense that ATPG should try to first merge the care bits for the second clock gate as it is just as effective as the first clock gate, and requiring much fewer care bits. While further analysis needs to be done, early results for design A were almost identical for the heuristics (a) and (b). When the clock gates were not sorted, patterns with repeat-fill show a slight increase in switching activity, while there is no such impact in the patterns generated with random-fill.

In the above experiments, all the flops were given equal weight. But in reality, some flops in the design have a higher impact on switching activity compared to other flops in the design. Instead of simply considering the number of flops controlled by a clock gate, some kind of a *weighted metric* could be considered as well. One simple *weight* to consider could be to count the number of gates in the fanout of a flop.

7. Conclusions

This paper presents an ATPG tool enhancement called CG-ATPG that limits the instantaneous power during the capture phase to the functional constraints. Some of the key contributions presented are:

1) Utilizes a comprehensive analysis of the hierarchical clock network, clock gating circuitry and granularity of flop gating control to selectively hold the value of some scan flip-flops during the capture cycle of a test pattern.
2) Addresses capture power reduction without relying exclusively on X-fill techniques that lead to the "saw tooth" switching problem described in [16].
3) Only the clock gates on the clock(s) being pulsed in the test pattern are considered. After enough clock gates have been turned off to bring the switching below a user specified threshold, the remaining don't-cares in the test pattern are filled using intelligent X-fill techniques to ensure that the load and unload shift power is also minimized.
4) Results are presented on three different industrial LP designs that demonstrate up to 30% (55%) reduction in instantaneous (average) capture switching.
5) For these designs, the increase in pattern count for the CG-ATPG technique was mostly within 10%, while the runtime increase was mostly within 15%.
6) CG-ATPG also helps identify ungated clock and asynchronous signals that contribute to high capture power and need to be addressed by enhancements to the tools that implement high fanout and clock tree networks.

Some on-going and future work will consider physical design information such as power grids and region-based power constraints similar to the work proposed in [20].

8. Acknowledgements

The authors would like to acknowledge and thank Richard Illman, Leon Palmer, James Youngs, Louis Milano, David Scott and Rajesh Khurana for their contributions to this work.

References

[1] P. Girard, "Survey of Low-power testing of VLSI Circuits," *IEEE Design and Test*, vol. 19, issue 3, May-June 2002, pp. 80-90.

[2] 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. Int. Test Conference*, October 2004, pp. 355-364.

[3] 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. Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, issue 7, July 2004, pp. 1142–1153.

[4] O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu, "Test power reduction through minimization of scan chain transitions," *Proc. VLSI Test Symposium*, May 2002, pp. 166-171.

[5] R. Sankaralingam, B. Pouya, and N. A. Touba, "Reducing power dissipation during test using scan chain disable," *Proc. VLSI Test Symposium*, May 2001, pp. 319-324.

[6] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A gated clock scheme for low power scan testing of logic ICs or embedded cores," *Proc. Asian Test Symposium,* 2001, pp. 253-258.

[7] X. Wen, *et. al.,* "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing," *Proc. VLSI Test Symposium*, May 2006, pp. 58–63.

[8] N. Nicolici, X. Wen, "Embedded Tutorial on Low Power Test*," Proc. 12th European Test Symposisum,* May 2007, pp. 202 – 210.

[9] ____, Si2 Common Power Formal Specifications, `http://www.si2.org/`

[10] ____"Unified Power Format (UPF) Standard, Version 1.0",`http://www.accellera.org/activities/p1801_upf/`

[11] Chickermane V, Gallagher P, Sage J, Yuan P, Chakravadhanula K, "A Power-Aware Test Methodology for Multi-Supply Multi-Voltage Designs," *Proc. Int. Test Conference*, Oct 2008, paper 9.1.

[12] Czysz D, Kassab M, Lin X, Mrugalski G, Rajski J and Tyszer J, "Low Power Scan Shift and Capture in the EDT Environment," *Proc. Int. Test Conference*, Oct 2008, paper 13.2.

[13] Furukawa H, Wen X, Yamato, Y, Kajihara S, Girard P, Wang LT, Tehranipoor M, "CTX: A Clock-Gating-Based Test Relaxation and X-Filling Scheme for Reducing Yield Loss Risk in At-Speed Scan Testing," *Proc. Asian Test Symposium*, November 2008, pp 397-402.

[14] Keller B, Scott D, Bhatia S, Illman R, "Low Power ATPG," *Proc. Cadence Technical Conference*, May 2007, paper 2C.2.

[15] Illman R, Keller B, Bhatia S, "A Review of Power Strategies for DFT and ATPG," *12th IEEE European Test Symposium*, May 2007, paper 2C.1.

[16] Illman R, Keller B, Gallagher P, "ATPG power reduction using clock gate 'default' constraints," *Proc. 1st International Workshop on the Implications of Low Power design on Test and Reliability*, 2008, pp.45-46.

[17] Remersaro S, Lin X, Zhang Z, Reddy SM, Pomeranz I, Rajski J, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs." *Proc. Int. Test Conference*, October 2006, paper 32.2.

[18] Wen X, Miyase K, Kajihara S, Suzuki T, Yamato Y, Girard P, Ohsumi Y, Wang LT, "A novel scheme to reduce power supply noise for high-quality at-speed scan testing," *Proc. Int. Test Conference*, October 2007, paper 25.1.

[19] Keller B and Chakravadhanula K, chapter "Test Strategies for Gated Clock Designs" in "Power-Aware Testing and Test Strategies for Low-Power Devices", P. Girard, N. Nicolici, and X. Wen (editors), ISBN: 978-1-4419-0927-5, Springer, New York, 2009.

[20] Devanathan V.R., Ravikumar C.P., Kamakoti V., **"**Glitch-Aware Pattern Generation and Optimization Framework for Power-Safe Scan Test," *Proc.VLSI Test Symposium*, May 2007, pp. 167-172.