Peak Capture Power Reduction for Compact Test Sets Using Opt-Justification-Fill

Stephan Eggersglüß*†
* Institute of Computer Science, University of Bremen, Germany
†German Research Center for Artificial Intelligence (DFKI), Bremen - Cyber-Physical Systems
{segg@informatik.uni-bremen.de}

***Abstract*—Excessive test power consumption is one of the obstacles which the chip industry currently faces. Peak capture power reduction typically leads to high pattern counts which increase test costs. This paper proposes a new methodology to reduce peak capture power during at-speed scan testing. In this method, a novel X-filling technique *Opt-Justification-fill* which uses optimization techniques to compute promising X-bits for low-power filling is proposed. This method is tightly integrated into a dynamic compaction flow to create silent test cubes with high compaction ability. By this, X-filling for fault detection and reducing switching activity is balanced. The proposed methodology can be applied during initial compact test set generation as well as a post-ATPG stage for a previously generated test set to reduce switching activity. Experiments show a significant reduction of peak capture power. At the same time, the pattern count increases only moderately which leads to reduced test costs.**

I. Introduction

At-speed scan testing is widely adopted to detect timing-related defects in manufactured devices. However, this kind of testing is challenged by the problem of test power consumption. The gap between functional power and test power is widening with the common practice of low power design techniques and increasing test power. The difference is reported to grow to up to a factor of 5 [1]. This may lead to unnecessary heat-related yield loss since thermal effects are able to invalidate the correct test responses. Therefore, dedicated test techniques are required to decrease the test power consumption.

The power consumption is typically measured in two different categories: shift power and capture power [2]. This paper focuses on the reduction of capture power in a launch-on-capture scan scheme. At-speed scan testing requires the consideration of two clock cycles named launch cycle (C_1) and capture cycle (C_2). Launching the test stimulus in C_1 may lead to excessive *Launch Switching Activity* (LSA) which, in turn, causes IR-drop. This may lead to timing failures in C_2 [3] which would not occur in functional mode. As reported in [4], the reduction of LSA is considered more important than the reduction of capture switching activity in C_2. The reduction of peak LSA is particularly important since the likeliness of a discrepancy between the test behavior and the intended functional behavior is highest here. Complementary to peak capture power reduction is the generation of launch-safe tests [5]. For the generation of launch-safe tests, the LSA impact of neighboring areas is considered.

Generally, capture power reduction is achieved by modified ATPG procedures [6], [7] or X-filling methods [8]–[14]. The reduction of capture power in critical areas is considered in [5], [15], [16]. Additionally, layout-aware techniques were proposed to increase the accuracy of the switching activity estimation targeting critical areas [17], [18]. A serious issue for capture power reduction is the accompanying test inflation. Typically, capture power reduction goes along with a significantly increasing pattern count which raises the test costs.

Since regular ATPG procedures are highly optimized, modified ATPG procedures suffer from the circumstance, that the optimized ATPG data structures and conflict detection mechanisms are compromised leading to a run time problem. On the other hand, X-filling techniques are ATPG-independent and use remaining X-bits in the generated test pattern which are not used for fault detection. Therefore, the ATPG engine has not to be modified and retains its efficiency.

A. Related Prior Work

The *Weighted Switching Activity* (WSA) is typically used as a metric to measure and reduce power consumption in an early design stage. The WSA metric of a node under test application is defined as the number of state changes at this node multiplied by $(1+N)$ where N is the number of fanouts. The WSA of the circuit is the sum of the WSA of all nodes [4].

An ATPG procedure to reduce capture power is proposed in [7]. Here, a modified PODEM algorithm is used to detect additional C-conflicts if scan flipflops switch during test generation. However, the ATPG modification leads to severe run time issues.

X-filling methods are often used, since these methods can easily be integrated into an existing test generation flow. In a conventional ATPG flow, test cubes are generated including a certain amount of don't care bits. These X-bits have to be eventually specified to 0 or 1. Random-Fill methods are typically used in order to increase the detection ability leading to a more compact test set. However, this kind of method is disadvantageous for obtaining a low switching activity. Instead, several X-filling methods were proposed to reduce the peak or average switching activity.

PMF-filling [9] fills the X-bits in the launch stimulus such that the Hamming distance between launch and capture stimulus is reduced. If the corresponding bits are both unspecified, random values are used. Preferred fill [10], [11] is a scalable

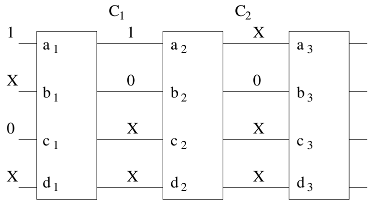

Fig. 1. Example - DC-fill

and effective X-filling method which extends this concept and uses signal probabilities to prevent random assignments. Instead, preferred values are used. The method proposed in [14] simulates the stimuli over multiple clock cycles and fills values which mimic functional operation.

Justification-based X-filling is conducted in [13] (DC-fill). Figure 1 shows a short example to illustrate the procedure. Four 3-bit couples are extracted from the input assignments of a test cube:

$$< a_1, a_2, X >, < X, b_2, b_3 >, < c_1, X, X >, < X, X, X >$$

One couple represents the input assignment of a (pseudo-) primary input over multiple clock cycles. The X-values are heuristically filled in an iterative way by means of justification such that the switching activity in the combinational logic is reduced. The heuristic calculates the justification easiness of the logic values to order the targets. However, the optimal filling combination might be missed since the justification is done independently for each logic value and logic dependencies might interfere. Additionally, the procedure is more time-consuming compared to probabilistic filling methods. Therefore, JP-fill [12] was proposed to balance the filling procedure by combining probabilistic methods and justification to reduce the number of justification calls.

The previous X-filling methods have in common that their methodology focuses on using X-bits for WSA reduction but consider compaction aspects rather secondary or not at all. This is addressed in this work.

Recently, SAT-based and optimization methods based on efficient SAT solving engines attracted attention in the field of test generation [19]–[23], i.e. by reducing aborts, maximizing fault detection and power-droop conditions. These methods are shown to provide good opportunities to solve problems with large search space. Therefore, optimization methods are used in this paper.

B. Contribution

This paper proposes a new Low-LSA ATPG method to generate “silent” test cubes using efficient optimization techniques. Unlike previous approaches, partial X-filling is conducted after *each* test cube generation or additional fault detection during a dynamic compaction flow before the test is finished. Only few relevant bits are filled using the proposed X-filling method *Opt-Justification-fill* (OJ-fill) leaving other X-values to be processed for detecting further faults. This procedure is able to influence further fault detection moderating the pattern count increase.

The ATPG formulation includes the sequential behavior using the launch-on-capture scan scheme. After a test cube is generated, justification targets with X-values at C_2 are extracted from the obtained test cube. An optimization procedure is applied in order to assign the justification targets to minimized switching activity. Partial X-filling is then conducted according to the computed necessary bit assignments. In contrast to previous justification-based X-filling methods, the optimization procedure processes all justification targets at once. By this, logic dependencies are taken into account and X-filling is performed using optimal values.

The primary ATPG flow is complemented by a post-ATPG process (for the complete test set) which substitutes peak LSA patterns by newly generated patterns with low switching activity. The experiments show that the proposed ATPG method is able to decrease peak LSA effectively. At the same time, the pattern count increases only moderately.

The paper is structured as follows. Section II briefly describes the formal methods used in this paper. Section III presents the proposed OJ-fill method while Section IV shows how the X-filling method is integrated into a dynamic compaction flow as well as how the post-ATPG process is conducted. Experiments are given in Section V and Section VI concludes the paper and presents future work.

II. Preliminaries

This section reviews briefly the use of *Boolean Satisfiability* (SAT) and *Pseudo-Boolean Optimization* (PBO) since these problems/methods are used in this work. The SAT problem is defined as the question whether there exists an assignment a to a Boolean formula f represented in *Conjunctive Normal Form* (CNF) such that f is satisfiable (SAT). If there is no such assignment, the formula is unsatisfiable (UNSAT). A CNF formula Φ is a conjunction of clauses and a clause ω is a disjunction of literals. A literal λ is a Boolean variable in its positive (λ) or negative ($\overline{\lambda}$) form.

SAT solvers have become very robust in solving CNF formulae due to the development of efficient implication techniques and effective conflict-based learning. This makes them very suitable for the application to problems with a large search space.

The PBO problem consists of a set of pseudo-Boolean constraints Ψ and an optimization formula F. The solution to the PBO problem is the assignment which satisfies Ψ and, at the same time, optimizes F. The optimization function is defined as follows:

$$F(x_0, \ldots, x_{n-1}) = \sum_{i=0}^{n-1} c_i x_i,$$

where $x_0, \ldots, x_{n-1} \in \mathbb{B}$ and $c_0, \ldots, c_{n-1} \in \mathbb{Z}$. In other words, the evaluation of F results in the accumulation of all constants c_i for which the corresponding Boolean variable x_i is assigned to 1.

Since a CNF can easily be transformed into a set of pseudo-Boolean constraints (but not vice versa), PBO solvers typically are able to process a CNF in a native way. In fact, most PBO solvers apply SAT solving engines to solve PBO problems if the original problem is CNF-based.

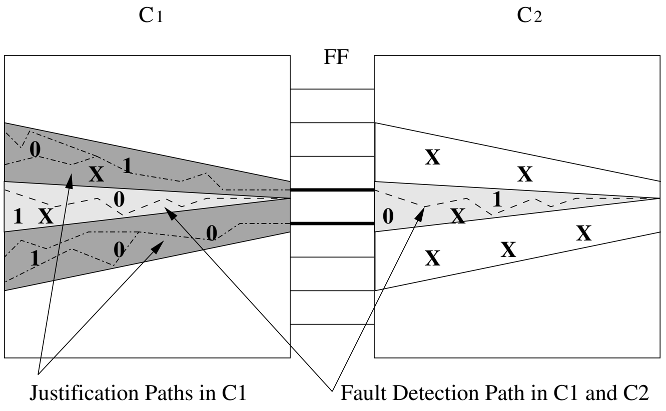

Fig. 2. Illustration of Fault Detection and Justification Paths

III. Opt-Justification Fill

This section introduces the novel *Opt-Justification-fill* method. Its application in the test generation flow will be described in Section IV. In previous methods, e.g. DC-fill or JP-fill, X-filling is conducted to a finalized test cube which is not processed by ATPG anymore. That is, either test vectors of a pre-computed test set are modified by test relaxation techniques [24], or the detection of additional faults is stopped during ATPG if a certain bound is reached. The result of this methodology is that X-filling cannot influence the additional fault detection mechanism anymore.

The difficulty of an earlier integration of X-filling is that X-bits have to be carefully chosen so that they are able to reduce switching activity at an early stage as well as do not prevent the ATPG from detecting additional faults. This problem is tackled by the proposed OJ-fill scheme.

The application of the OJ-fill scheme is motivated by an analysis of test cube generation and their fault detection and justification paths. This is illustrated in Figure 2.

If a fault detection path p_f for a target fault f is found by ATPG, certain necessary values at the side inputs of p_f had to be justified by the ATPG such that a (partial) input assignment implies these values. For at-speed testing, the sequential behavior has to be considered. Therefore, the assignment of pseudo-primary inputs, i.e. flipflops, in C_2 is not sufficient. These values assigned in C_2 had to be justified in C_1 as well. Consequently, each flipflop assignment in C_2 corresponds to one or more justification paths in C_1 as shown in Figure 2. These justification paths establish the necessary value assignments at the flipflops.

Preliminary experiments showed that the number of X-values is much higher in C_2 than in C_1 since more logic has be considered in C_1 due to the flipflop justification during ATPG. This results in a large portion of potential switching activity if these X-values are used to detect other additional faults by setting them to the opposite value.

The aim of the proposed OJ-fill method is to justify these X-values such that the potential switching activity is eliminated to produce a “silent” test cube. At the same time, only few X-values are to be specified in order to retain the possibility to detect additional faults by ATPG. In order to achieve this goal, justification targets are to be extracted and to be suitable assigned. This is explained in the following.

A. Justification Target Extraction

A justification target is a signal line in the combinational logic which is partly assigned to an X-value and, therefore, can potentially produce switching activity. Justification targets in C_2 are extracted based on justification paths in C_1. A justification path p_j is defined as a structural path $s_1, \ldots, s_n$ from an input s_1 to an end line s_n whose value assignment is necessary to imply the value of s_n. Here, the end line corresponds to an output in C_1 whose value is needed in C_2 for fault detection. Please note that there are typically more justification paths necessary to imply the value of s_n. The set of lines on all justification paths for end line s_n is called a *justification line set* in the following. The justification line set can be extracted by path tracing or test relaxation techniques [24]. Using test relaxation is advantageous because unnecessary specified bits can be unspecified during this procedure.

After a test cube t has been generated and simulated, the following procedure is used:

1) Extract the set of assignments A of the justification line set in C_1 in the following form:
$$(s_1^1, v_1), \ldots, (s_n^1, v_n)$$
where $s_1^1, \ldots, s_n^1$ are the signal lines in C_1 and $v_1, \ldots, v_n$ the assigned Boolean values in C_1.
2) Generate the set of justification targets J out of the extracted assignments A, i.e. replace each line in C_1 by its corresponding part in C_2 and keep the logic value:
$$(s_1^1, v_1), \ldots, (s_n^1, v_n) \rightarrow (s_1^2, v_1), \ldots, (s_n^2, v_n)$$
where s_i^2 is the corresponding line in C_2 of line s_i^1 in C_1.
3) Remove all satisfied or unsatisfiable justification targets from J, i.e. delete all justification targets whose signal lines have values other than X in C_2 under the application of t.

The application of this procedure results in a set J of justification targets in C_2 which are assigned to X.

B. Optimization-based X-Filling

Given the set of justification targets J, the aim is to satisfy as many targets as possible at the same time to reduce potential switching activity. The justification of all targets might not be possible due to logic correlations between the targets. Therefore, an optimization procedure is applied to compute the logic assignment which satisfies the maximum number of targets.

A PBO procedure is applied as follows:

1) Extract the logic cone of all justification targets spanned over C_2 and C_1 as illustrated in Figure 3.
2) Generate the SAT instance Φ_J of the extracted logic cones.
3) Set all inputs with specified values in t included in Φ_J to its corresponding value.

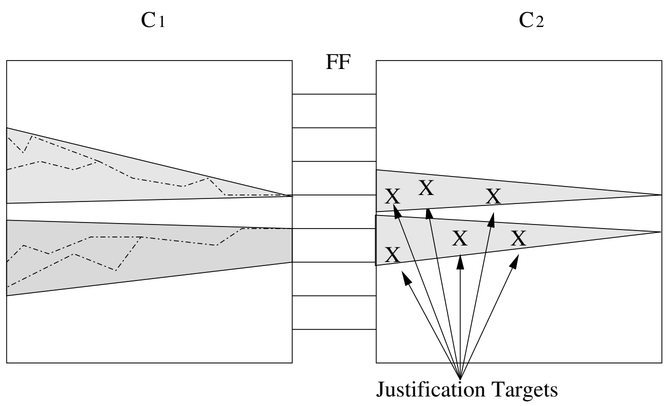

Fig. 3. Logic Cone Extraction

4) Create the optimization function F_J based on the justification targets J:

$$F(s_1, \ldots, s_n) = \sum_{i=1}^{n} c_{\text{WSA}} \cdot \dot{s}_i \begin{cases} \dot{s}_i = \overline{s}_i, & \text{if } v_i = 0. \\ \dot{s}_i = s_i, & \text{otherwise.} \end{cases}$$

The constant c_{WSA} is the local WSA of the signal line in order to prioritize fanouts with more branches, since it is advantageous to eliminate switching activity there.

5) Solve the SAT instance Φ_J in conjunction with F_J by a PBO solver. The result is a complete assignment a of the logic cones satisfying the maximum number of targets.

The logic assignment can be used to fill X-bits in the test cube t. However, filling all inputs of the considered logic cones would result in a significant decrease of X-bits. By this, significantly less bits can be assigned to detect additional faults. Therefore, test relaxation is applied to extract the justification paths of all justification targets. The inputs of all justification paths are then filled by their computed values.

As a result of this optimization-based X-filling, a "silent" test cube is produced which still has many X-bits for detecting additional faults.

IV. Dynamic Compaction Flow

A disadvantage of previous justification X-filling methods is that they are not able to influence the potential detection of additional faults. They are applied to fill the remaining X-bits after a test cube has been finished. This means, decisions made by the ATPG can hardly be changed even if they are disadvantageous for switching activity reduction, although a different "better" way of detecting this fault might be possible.

In a dynamic compaction flow, a test cube t_p for a primary target fault f_p is generated first. After that, secondary target faults f_s are chosen and the ATPG is called to generate a test cube t_s under the assumption that all input assignments of t_p are maintained. In other words, the ATPG tries to fill the remaining X-bits of t_p in a way that f_s is additionally detected.

We propose an intermediate stage in which the OJ-fill method is applied to "silence" the generated test cube. The additionally filled bits can then be used by the subsequent ATPG call to influence the search for additional faults. By this, the ATPG can be indirectly guided to detect faults with potentially low switching activity without modifying the ATPG core engine itself.

Algorithm 1 Dynamic Compaction Flow Using OJ-Fill

```
 1: TestSet $T$ = $\emptyset$;
 2: FaultList $F_\text{p}$ = AllUndetectedFaults();
 3: while $F_\text{p} \neq \emptyset$ do
 4:     Fault $f_\text{p}$ = PopFault($F_\text{p}$);
 5:     Test cube $t_p$ = DoATPG($f_\text{p}$);
 6:     if $t_p == \emptyset$ then
 7:         continue;
 8:     end if
 9:     $t_p$ = DoOJ-fill($t_p$);
10:     FaultList $F_\text{s}$ = AllUndetectedFaults();
11:     while $F_\text{s} \neq \emptyset$ do
12:         Fault $f_\text{s}$ = PopFault($F_\text{s}$);
13:         $t_s$ = DoATPG($f_\text{s}$, $t_p$);
14:         if $t_s == t_p$ then
15:             continue;
16:         end if
17:         $t_s$ = doOJ-fill($t_s$);
18:         $t_p = t_s$
19:     end while
20:     $T = T \cup t_p$;
21:     $F_\text{p}$ = AllUndetectedFaults();
22: end while
23: return $T$;
```

The modification of the dynamic compaction flow remains small. The dynamic compaction flow which is used is shown in Algorithm 1. The only extension are the OJ-fill calls in line 9 and line 17. After a test cube for a primary fault f_p is generated, OJ-fill is conducted to fill some of the X-bits to eliminate potential switching activity. After that, the inner loop is entered to target secondary faults f_s to fill X-bits to detect additional faults. OJ-fill is then used to fill further X-bits for each additional detected secondary fault.

Note that after the secondary target loop has been finished, all remaining X-bits are filled by a probabilistic-based X-filling method such as Preferred fill.

The proposed flow extension makes sure that X-filling for reducing switching activity (by OJ-fill) as well as X-filling for detecting additional faults (by ATPG) are balanced to achieve a good trade-off between additional fault detection and reduced switching activity.

A. Post-ATPG Stage

Additionally to the proposed dynamic compaction flow to generate an initial test set, the methodology can also be used to improve an already generated test set. Similar to the Low-LSA ATPG proposed in [7], a post-ATPG stage is used to replace high-capture power tests by low-capture power tests to reduce the peak capture power. Instead of replacing one test at a time by one or more tests, multiple tests are replaced in one step. The following procedure has been developed in order to improve the test set.

1) Calculate the WSA in C_1 and C_2 of all tests in T.
2) Order all tests $t_1, t_2, \ldots, t_n \in T$ such that

$$\text{WSA}_{C_1}(t_1) \geq \text{WSA}_{C_1}(t_2) \geq \ldots \geq \text{WSA}_{C_1}(t_n)$$

3) Remove the first $p\%$ tests from T, i.e. those with the peak WSA. Experiments have shown that $p = 7$ provides good results.
4) Set all faults which are not detected by T anymore to an undetected fault state.
5) Retarget all faults which are undetected by T by the dynamic compaction flow presented above with a reverse fault ordering. Add the newly generated tests to T.
6) Proceed with 1) until sufficient improvement is achieved or the test set size grows over a user-specified limit.

The advantage of using a post-ATPG stage is that only a small set of faults is considered at a time. Since the fault processing order is changed, other justification targets are chosen leading to different X-filling results. By this, slightly more tests are generated in each step but with typically reduced switching activity. However, a peak reduction is not guaranteed in each step, but the experiments in the next section show that the OJ-fill method is mostly able to cause a sufficient diversity in X-bit assignments to prevent high switching activity.

V. Experimental Results

This section presents the experimental results of the proposed approach. The approach was implemented in C++ and experiments were conducted on a Intel Xeon E3 (GNU/Linux, 3.4GHz). The techniques were integrated into an in-house academic ATPG environment. The PBO solver clasp [25] was used to solve the optimization problems.

Table I gives details about some statistics and the overhead in terms of run time and pattern count increase. Additionally, the overhead in run time and pattern count is compared to the data given in [11] (Preferred fill, PF) and [13] (DC-fill).

Results for the proposed approach are given in three different configurations:

- *Initial ATPG* – The dynamic compaction flow using OJ-fill (without the post-ATPG stage) is applied to generate a compact test set with low peak switching activity.
- *Post-ATPG* – The proposed post-ATPG procedure using OJ-fill is applied to a previously generated test set using Random-fill in order to reduce switching activity.
- *Initial + Post* – First, the dynamic compaction flow using OJ-fill is applied. After the test set has been generated, the post-ATPG stage is used to decrease peak switching activity further.

Columns *RF-Pat* gives the number of patterns of the test set generated with random fill to detect additional faults. Columns *Pat* gives the increase of the pattern count compared to the corresponding RF-Pat number. Column *Time* shows the run time overhead. For the initial ATPG, the run time overhead is given as a factor compared to test generation with random fill. For the post-ATPG stage, the time is given in total CPU seconds. Note that the run time of the initial ATPG have to be added for the combination. The columns *OJ-bits* presents the average portion of bits set by the OJ-fill method.

Table II presents the reduction of the peak switching activity.[1] Again, the results of the proposed method are compared to PF and DC-fill. The results are split into the first capture (LSA) and the second capture reduction. Results for the proposed approach are given in the three configurations described above. The disadvantage of the proposed techniques is that the peak switching reduction in the second capture cycle is not as much reduced as using PF or DC. Nonetheless, a certain amount of reduction can be achieved. However, the reduction of LSA is considered more important as reported in [4].

Although, the run time increases by 1.52X on average, the results show that the proposed initial dynamic compaction flow using OJ-fill is very effective in achieving a good reduction of peak LSA with a marginal pattern count overhead compared to the high number of patterns produced by PF or DC. The pattern count increases only by 4.4% on average. Even less pattern than before are needed for three circuits. However, the peak LSA reduction is less than using PF or DC. But these methods need significantly more tests on average. The small pattern overhead can be explained by the number of OJ-bits. Only 14% selected bits are set using OJ-fill. PF- and DC-filling reserve 20% of the total bits for X-filling.

The post-ATPG stage is able to achieve a significantly higher peak LSA reduction. The reduction is also much higher than the reduction achieved by PF or DC. At the same time, the pattern count increases only very moderate on average. The pattern counts are higher than for the initial ATPG but still much lower that the pattern counts of PF and DC. The run time of the post-ATPG stage is scalable and acceptable.

The highest peak LSA reduction can be achieved by the combination of the inital ATPG procedure with the post-ATPG stage with up to 52.3% of reduction and with an average reduction of 37.7% which is a significant achievement. The pattern count is slightly lower than obtained by using post-ATPG only with 16.1% increase on average. This is far lower than the pattern count increase of PF and DC. The proposed techniques produce a very balanced test set with higher peak LSA reduction and lower pattern counts than the previous approaches PF and DC.

In summary, the proposed approach is able to reduce peak LSA very effectively and, at the same time, produces only a moderate test inflation which make the proposed method very cost-effective.

VI. Conclusions

The new X-filling method *Opt-Justification-Fill* (OJ-fill) and its integration into a dynamic test compaction flow have been presented. The integration of the X-filling method leads to a good balance of using X-bits for detecting additional faults and for reducing switching activity. The method can be effectively applied during initial compact test set generation to reduce peak *Launch Switching Activity* (LSA) as well as during a post-ATPG stage to further reduce peak LSA of pre-generated test sets.

Experimental results show that a significantly higher reduction of peak LSA can be achieved compared to previous

[1] The approach also reduces average switching activity in a similar way. The obtained results are not presented in this paper due to page limitation.

TABLE I
EXPERIMENTAL RESULTS – RUN TIME AND PATTERN COUNT

	Previous Work				Proposed Approach										
	PF [11]		DC [13]			Initial ATPG			Post-ATPG			Initial + Post			
Circ	RF-Pat	Pat	RF-Pat	Pat	RF-Pat	Time	OJ-bits	Pat	Time	OJ-bits	Pat	Time	OJ-bits	Pat
s1423	82	36.6%	132	13.6%	84	2.10X	17.0%	11.9%	2.6s	15.0%	24.4%	2.7s	15.5%	23.8%
s5378	167	23.4%	185	45.9%	117	1.59X	15.7%	16.2%	5.0s	15.0%	27.4%	1.5s	15.3%	23.9%
s9234	328	19.5%	483	26.1%	323	1.37X	9.5%	0.9%	17.2s	8.5%	1.5%	28.1s	10.2%	4.7%
s13207	377	2.1%	324	11.7%	424	1.04X	8.8%	-2.6%	26.1s	10.5%	2.8%	12.7s	10.0%	0.9%
s15850	183	10.4%	238	9.2%	168	1.27X	11.2%	-1.2%	15.7s	10.6%	11.3%	13.4s	12.1%	1.8%
s35932	40	140.0%	254	78.7%	43	2.06X	21.4%	-9.3%	8.8s	11.2%	23.3%	5.3s	15.3%	11.6%
s38417	222	45.9%	384	21.1%	164	1.34X	9.0%	10.4%	106.6s	7.6%	39.0%	112.5s	8.6%	45.7%
s38584	292	9.6%	444	9.9%	388	1.38X	19.3%	8.5%	66.9s	19.1%	18.0%	19.8s	19.7%	16.5%
Average	211	35.9%	306	27.0%	214	1.52X	14.0%	4.4%	31.1s	12.2%	18.5%	24.5s	13.3%	16.1%

TABLE II
EXPERIMENTAL RESULTS – PEAK SWITCHING ACTIVITY REDUCTION

	LSA – First Capture					Second Capture				
Circ	PF [11]	DC [13]	Initial ATPG	Post-ATPG	Initial + Post	PF [11]	DC [13]	Initial ATPG	Post-ATPG	Initial + Post
s1423	40.8%	-1.6%	21.4%	49.1%	38.6%	16.7%	4.5%	1.2%	8.0%	1.2%
s5378	30.1%	30.7%	35.6%	45.8%	48.0%	26.3%	33.3%	11.3%	7.5%	11.3%
s9234	19.1%	20.1%	24.6%	31.9%	33.0%	-1.1%	13.0%	10.6%	2.5%	10.6%
s13207	26.6%	22.0%	14.1%	22.5%	29.6%	22.8%	19.9%	5.7%	3.6%	12.9%
s15850	35.6%	24.9%	19.9%	34.2%	37.4%	21.9%	30.6%	10.9%	10.8%	11.9%
s35932	23.5%	22.4%	22.1%	31.0%	35.6%	11.8%	4.3%	5.8%	8.8%	10.9%
s38417	29.2%	30.2%	13.3%	27.6%	26.7%	21.9%	23.7%	10.8%	17.6%	21.5%
s38584	27.6%	50.3%	28.4%	48.6%	52.3%	35.7%	55.0%	16.1%	27.5%	30.1%
Average	29.1%	24.9%	22.4%	36.3%	37.7%	19.5%	23.0%	9.1%	10.8%	13.8%

methods. At the same time, the pattern count is much lower which leads to a test cost reduction.

Future work is the application of OJ-fill to high-quality test generation where the problem of test inflation is even more serious. Additionally, the accuracy of peak capture power is to be improved by integrating information about the layout or critical areas that peak capture power can be reduced in a more focused manner.

REFERENCES

[1] S. Sde-Paz and E. Salomon, "Frequency and power correlation between at-speed scan and functional tests," in *International Test Conference*, 2008, pp. 1–9.

[2] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 82–92, 2002.

[3] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, "A case study of IR-drop in structured at-speed testing," in *International Test Conference*, 2003, pp. 1098–1104.

[4] P. Girard, N. Nicolici, and X. Wen(Eds.), *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, 2009.

[5] X. Wen, K. Enokimoto, K. Miyase, Y. Yamato, M. A. Kochte, S. Kajihara, P. Girard, and M. Tehranipoor, "Power-aware test generation with guaranteed launch safety for at-speed scan testing," in *VLSI Test Symposium*, 2011, pp. 166–171.

[6] F. Corno, P. Prinetto, M. Rebaudengo, and M. S. Reorda, "A test pattern generation methodology for low power consumption," in *VLSI Test Symposium*, 1998, pp. 453–457.

[7] X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K. Saluja, L.-T. Wang, K. Abdel-Hafez, and K. Kinoshita, "A new ATPG method for efficient capture power reduction during scan testing," in *VLSI Test Symposium*, 2006, pp. 58–65.

[8] K. M. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis, and G. Hetherington, "Minimizing power consumption in scan testing: pattern generation and DFT techniques," in *International Test Conference*, 2004, pp. 355–364.

[9] W. Li, S. M. Reddy, and I. Pomeranz, "On reducing peak current and power during test," in *IEEE Annual Symposium on VLSI*, 2005, pp. 156–161.

[10] S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz, and J. Rajski, "Preferred fill: A scalable method to reduce capture power for scan based designs," in *International Test Conference*, 2006, pp. 1–10.

[11] S. Remersaro, X. Lin, S. M. Reddy, I. Pomeranz, and J. Rajski, "Scan-based tests with low switching activity," *IEEE Design & Test of Computers*, vol. 24, no. 3, pp. 268–275, 2007.

[12] 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," in *International Test Conference*, 2007, pp. 1–10.

[13] X. Wen, K. Miyase, T. Suzuki, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita, "Low capture switching activity test generation for reducing IR-drop in at-speed scan testing," *Journal of Electronic Testing: Theory and Applications*, vol. 24, no. 4, pp. 379–391, 2008.

[14] E. K. Moghaddam, J. Rajski, S. M. Reddy, and M. Kassab, "At-speed scan test with low switching activity," in *VLSI Test Symposium*, 2010, pp. 177–182.

[15] X. Wen, K. Miyase, T. Suzuki, S. Kajihara, Y. Ohsumi, and K. K. Saluja, "Critical-path-aware X-filling for effective IR-drop reduction in at-speed scan testing," in *Design Automation Conference*, 2007, pp. 527–532.

[16] K. Enokimoto, X. Wen, Y. Yamato, K. Miyase, H. Sone, S. Kajihara, M. Aso, and H. Furukawa, "CAT: a critical-area-targeted test set modification scheme for reducing launch switching activity in at-speed scan testing," in *IEEE Asian Test Symposium*, 2009, pp. 99–104.

[17] J. Lee and M. Tehranipoor, "Layout-aware transition-delay fault pattern generation with evenly distributed switching activity," *Journal of Low Power Electronics*, vol. 4, no. 3, pp. 1–12, 2008.

[18] ——, "LS-TDF: Low-switching transition delay fault pattern generation," in *VLSI Test Symposium*, 2008, pp. 227–232.

[19] R. Drechsler, S. Eggersglüß, G. Fey, A. Glowatz, F. Hapke, J. Schloeffel, and D. Tille, "On acceleration of SAT-based ATPG for industrial designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1329–1333, 2008.

[20] S. Eggersglüß, M. Yilmaz, and K. Chakrabarty, "Robust timing-aware test generation using pseudo-boolean optimization," in *IEEE Asian Test Symposium*, 2012, pp. 290–295.

[21] A. Czutro, M. Sauer, I. Polian, and B. Becker, "Multi-conditional SAT-ATPG for power-droop testing," in *IEEE European Test Symposium*, 2012, pp. 1–6.

[22] M. Sauer, S. Reimer, T. Schubert, I. Polian, and B. Becker, "Efficient SAT-based dynamic compaction and relaxation for longest sensitizable paths," in *Design, Automation and Test in Europe*, 2013, pp. 448–453.

[23] S. Eggersglüß, R. Wille, and R. Drechsler, "Improved SAT-based ATPG: More constraints, better compaction," in *International Conference on Computer-Aided Design*, 2013.

[24] K. Miyase and S. Kajihara, "XID: Don't care identification of test patterns for combinational circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 2, pp. 321–326, 2004.

[25] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub, "Conflict-driven answer set solving," in *International Joint Conference on Artificial Intelligence*, 2007, pp. 386–392.