A Two-Stage SAT-based ATPG Approach with Reduced Switching Activity

Stephan Eggersglüß Daniel Tille Rolf Drechsler
Institute of Computer Science, University of Bremen
Bibliothekstr. 1, 28359 Bremen, Germany
{segg, tille, drechsle}@informatik.uni-bremen.de

I. Introduction

Due to the rising performance and decreasing feature sizes of today's designs, the likelihood of delay defects increases. Therefore, transition fault testing is widely used to ensure that the delivered chips are free of fabrication defects. Due to the use of scan testing, the switching activity in the *Circuit Under Test* (CUT) is typically several times higher than during normal functional operation [1], [2]. This may lead to additional yield loss or even affect the reliability when "good" chips are exposed to excessive switching activity, which does not occur in normal operation mode. Common ATPG algorithms typically do not consider switching activity during test pattern generation for reasons of efficiency. However, in [3], a modified PODEM algorithm is applied to reduce the switching activity during test generation. In [4], power constraints are added to the circuit during test generation.

Recently, ATPG based on *Boolean Satisfiability* (SAT) has shown to be an efficient and robust alternative to classical ATPG algorithms [5]. A significant advantage of SAT-based algorithms is the powerful conflict analysis and the use of learned information.

In this paper, we present the concept of a SAT-based ATPG approach to generate tests with reduced switching activity between the launch cycle and the capture cycle. The approach consists of two consecutive stages. In the first stage, a normal transition fault test is generated. If the targeted fault is testable, the SAT instance is extended incrementally in the second stage to generate a test with reduced switching activity. In order to maintain a high fault coverage, a relaxation technique is described for the second stage.

II. SAT-based ATPG

State-of-the-art SAT solvers typically work on a Boolean formula represented in *Conjunctive Normal Form* (CNF). A CNF is a conjunction of clauses. A clause is a disjunction of literals and a literal is a Boolean variable in its positive form (x) or negative form ($\overline{x}$). To apply a SAT solver to a circuit-oriented problem, e.g. ATPG, the circuit has to be modeled in CNF. In the following, Φ_C corresponds to the CNF of circuit C. For generating a test for fault F, the fault specific constraints Φ_{FS} have to be added to Φ_C. Finally, information concerning the potential propagation paths, i.e. D-chains, – denoted by Φ_D – is added. Therefore, the CNF $\Phi_F = \Phi_C \wedge \Phi_{FS} \wedge \Phi_D$ describes the ATPG problem for some fault F in a circuit C. For more information on circuit-to-CNF transformation and fault modeling, we refer to [5].

In the following, we concentrate on transition faults. Evaluating the CNF Φ_F results in either untestability (if Φ_F is unsatisfiable) or in a test (if Φ_F is satisfiable). Here, the test is directly derived from the computed solution.

When generating delay tests, two time frames t_1, t_2 have to be considered. This is achieved by duplicating the circuit. Each copy represents one time frame. Consequently, a signal s is represented by two Boolean variables s^1, s^2, which describe the signal in the corresponding time frame. The behavior of the flip-flops is modeled by connections between both copies.

III. Encoding of Static Values

The representation of a signal with two Boolean values as used in common SAT-based ATPG approaches has the drawback that the absence of switching activity cannot be guaranteed. For example, consider a 2-input AND gate g. If both inputs switch in different directions, the Boolean representation causes that the output value of g remains 0 in t_1, t_2. This is because the controlling value 0 is assumed at one input in each time frame. The controlling value of a gate is the logic value, which, when assumed by one input, determines the output's value regardless from the value of other inputs. However, if the transitions on the inputs do not arrive simultaneously, a glitch is produced at the output which cannot be observed using the Boolean values.

For guaranteeing the absence of switching activity (transition and glitches) or manipulate the amount of switching activity, static values have to be encoded. This can be done by assigning a third variable s^s to signal s. The variable s^s determines whether the signal is static between t_1 and t_2 or not. Additional constraints have to be added to the SAT instance for the computation of static values. Given a gate g with inputs $i_1, \ldots, i_n$, the controlling value *cv* and the non-controlling value *ncv*. If g is static and assumes *ncv*, then $i_1, \ldots, i_n$ must be static, too. If g is static and assumes *cv*, then at least one input i_j with $1 \leq j \leq n$ must assume *cv*. formally, the following

These implications can be transformed into CNF and are denoted by Φ_S in the following. The CNF Φ_S is logically redundant, i.e. the solution space concerning the number of tests remains the same. However, due to the this formulation, the static signals can be identified and manipulated as shown in the next section. The advantage of this encoding is that it can be incrementally added to the SAT instance and thus, all learned information can be kept. The effectiveness of using this information in subsequent solving process was shown for example in [6].

IV. Two-Stage Approach

Here, the overall two-stage approach is described. The pseudo-code is presented in Algorithm 1. At first, transition fault test generation is performed as usual (first stage; lines 1-5). If the fault is untestable, the second stage is not entered. If the fault is testable, the second stage is started to find a test which is likely to have reduced switching activity. Therefore, the SAT instance Φ_F is augmented by the static value encoding Φ_S resulting in Φ_{SF} (line 7). As major benefit, the information learned so far during the search process is still valid and can therefore be applied in the second stage to exclude illegal signal value assignments.

To generate a test for fault F, the fault must be excited at the fault site and then be propagated to an observation point. The handling of the off-path inputs is crucial for the generation of a test with reduced switching activity. An off-path input is a signal that drives a gate on a propagation path, i.e. on a D-chain, but is itself not located on this path. Normally, the off-path inputs of the D-chain are only constrained to propagate the fault effect. But often, a large part of the combinational logic of the circuit is needed to justify the off-path inputs. Therefore, the goal is to increase the number of static off-paths inputs to reduce the overall switching activity.

This can be done by additional constraints. Consider a gate g with inputs $i_1, \ldots, i_n$. If, and only if, input i_j is on a D-chain, all other inputs have to assume static values to reduce the switching activity. The resulting implications in CNF are denoted by Φ_O. Then, the SAT instance consisting of Φ_{SF} and Φ_O is solved (including the learned information from the first stage; line 9). If the SAT instance is satisfiable, a test is found which is very likely to have reduced switching activity due to the static off-path inputs.

However, restricting the off-path inputs to static values results in a decreased fault coverage, because some faults may become untestable due to Φ_O. To keep the fault coverage high, a relaxation procedure is proposed for Φ_O. Because F was originally testable, the source of the unsatisfiability has to be in Φ_O. An *unsatisfiable core* [7] is therefore generated and used to pinpoint the source of the unsatisfiability. Afterwards, Φ_O can be relaxed according to the unsatisfiable core (line 11), i.e. some off-paths inputs do not have to assume static values. Then, the alternated SAT instance is solved again (lines 10-13) until a test is found. In the worst case, Φ_O is completely discarded. As an alternative to save run time, the test generated in the first stage can be directly used after the first iteration.

V. Preliminary Results

Table I shows the first results of a rough prototypical implementation for generating restricted broadside tests (without the relaxation technique). Here, the test from the first stage is taken if the second stage determines unsatisfiability. As measurement for switching activity, the number of switching gates was chosen. Column *Av. Sw.* shows the average number of switching gates for one test. The maximum number of switching gates per test is shown in column *Peak*. The average reduction of the number of switching gates is presented in column *Av. red. %*. The potential of the approach can clearly be seen. Here, the maximum switching activity is significantly reduced.

Algorithm 1 Pseudo-code of the two-stage approach

1: Select_Transition_Fault F;
2: Generate Φ_F;
3: Solve Φ_F;
4: **if** Φ_F = UNSAT **then**
5:     **return** UNTESTABLE
6: **else**
7:     $\Phi_{SF} = \Phi_F \wedge \Phi_S$;
8:     Generate Φ_O;
9:     Solve ($\Phi_{SF} \wedge \Phi_O$);
10:     **while** $\Phi_{SF} \wedge \Phi_o$ = UNSAT **do**
11:         Relax Φ_O;
12:         Solve $\Phi_{SF} \wedge \Phi_O$;
13:     **end while**
14: **end if**
15: **return** Test T;

TABLE I
Experimental Results - Number of switching gates

	Classic			Two-stage			
Circ	Av. Sw.	Peak	time	Av. Sw.	Peak	time	Av. red.
b04	122.7	375	0:02m	101.0	292	0:03m	10.9%
b13	26.1	99	0:01m	22.2	72	0:01m	8.1%
b14	1067.7	5380	3:20m	656.1	4050	5:26m	32.8%
b15	924.3	2816	4:42m	691.7	2291	7:11m	17.2%

Furthermore, the number of switching gates is decreased as well by up to 32.8%. At the same time, the run time overhead is moderate.

VI. Conclusions and Outlook

We presented the concept for a two-stage SAT-based ATPG approach for generating tests with reduced switching activity. Constraints for reducing switching activity are added in an incremental manner so that learned information from the first stage can be reused in the second stage to improve the efficiency. First results have shown the potential of this approach.

So far, the approach considers only off-path inputs of the propagation paths. Future work is the extension of the concept to the off-path inputs of the activation path. Furthermore, the approach will be evaluated with other measurements, e.g. weighted switching activity.

References

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

[2] N. Nicolici and B. Al-Hashimi, *Power-constrained testing of VLSI circuits*. Kluwer Academic Publishers, Boston, MA, 2003.

[3] 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 Symp.*, 2006, pp. 58–65.

[4] S. Ravi, V. R. Devanathan, and R. Parekhji, "Methodology for low power test pattern generation using activity threshold control logic," in *Int'l Conf. on Computer-Aided Design*, 2007, pp. 526–529.

[5] 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 Trans. on Computer-Aided Design for Circuits and Systems*, vol. 27, no. 7, pp. 1329–1333, 2008.

[6] S. Eggersglüß and R. Drechsler, "Increasing robustness of SAT-based delay test generation using efficient dynamic learning techniques," in *IEEE European Test Symp.*, 2009.

[7] L. Zhang and S. Malik, "Validating SAT solvers using an independent resolution-based checker: Practical implementations and other applications," in *Design, Automation and Test in Europe*, 2003, pp. 880–885.