ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

High Fault Coverage ATPG for Industrial Application using DYNAMIC SAT Technique

V.M.Thoulath Begam[1], DR. S.Baul Kani [2],

Assistant Professor, Department of Electronics and Communication Engineering, National College of Engineering, Tirunelveli, Tamil Nadu[1]

Associate Professor, Department of Electronics and Communication, Government College of Engineering, Tirunelveli, Tamil Nadu[2]

Abstract: It is a novel technique for automatic test pattern generation which well detects both easy to detect faults and hard to detect faults. ATPG based on this SAT technique dynamic clause activation (DCA) generates a limited number of test patterns which can cover more faults. ATPG based on implication graph have problems to cope with hard-to-detect faults. ATPG based on Boolean satisfiability does not work on a structural form. In SAT solver method the problem is transferred in to a Boolean formula and a SAT solver is used to solve this problem. It has disadvantages such as overhead for CNF transformation and over specified solutions. In the proposed method, the solving problem is directed by structural information which is provided dedicated data structures. The technique is designed such that the efficient solving techniques and data structures of SAT solver do not have to be modified. This is a crucial to retain the high efficiency of a SAT based algorithm. The proposed method is fast and provides a very high fault efficiency and useful in large industrial circuit.

Keywords: ATPG, Boolean Satisfiability, Dynamic SAT, Fault Coverage, SA technique.

I. INTRODUCTION

Each manufactured chip has to be subjected to a postproduction test in order to filter out defective devices. Test patterns are applied which are generated by algorithms for automatic test pattern generation (ATPG). Classical ATPG algorithms such as FAN [1], which are widely used in industry, work on a structural net list or on an implication graph. These algorithms are very fast and many faults can typically be classified very quickly. Although being heavily improved in the last decades, e.g., by [2], these structural algorithms reach their limits and have problems to cope with hard-to-detect faults whose number is steadily increasing in today's complex designs. This leads to a growing proportion of faults which cannot be classified—especially for the transition fault model. As a result, there is a renewed interest of the industry in novel efficient ATPG algorithms. ATPG based on Boolean satisfiability (SAT) was proposed in the 1990s [3]. SAT-based ATPG does not work on a structural net list but on a Boolean formula in conjunctive normal form (CNF). The problem is transformed into a Boolean formula _ and a SAT solver, e.g., [8], is used to solve this formula. However, the early approaches did not become widely accepted because of some disadvantages such as overhead for CNF transformation, and over specified solutions. Additionally, existing structural ATPG algorithms were fast enough to cope with designs of that time.

However, the efficiency of SAT solvers has been significantly improved in the last decade and recent SAT-based ATPG approaches [4]-[7] turned out to be an efficient complement to existing classical ATPG algorithms classifying many faults for which structural algorithms determine no solution in reasonable time. Although being very robust, they suffer from the overhead for easy-to-detect faults caused by the dominating CNF transformation time, the loss of structural knowledge, and a high number of specified bits. Since easy-to-detect faults typically represent the majority of all faults, this leads to unacceptable high ATPG run times for some industrial circuits which makes the stand-alone application of SAT-based ATPG in industrial practice unfeasible. Hybrid SAT solvers work—at least partly—on the problem in CNF representation but exploit structural knowledge to speed up the search or use structural implication procedures, respectively. The approach presented in [9] retains the gate connectivity information of the circuit as also done in the proposed approach. In proposed incremental SAT technique dynamic clause activation (DCA) which has been specially developed for the application on ATPG problems. It is the first ATPG technique that integrates a (dynamic) SAT instance generation within the solving process. The solving process is directed by structural information which is provided by dedicated data structures. The technique is designed such that the efficient solving techniques and data structures of a SAT solver do not have to be modified. This is crucial to retain the high efficiency of a SAT-based algorithm. In addition, this leads to a removal of the limitations of classical SAT based ATPG which prevented the use in industrial practice. This paper is organized as follows. Basic information about SAT-based ATPG is given in Section II. Section III presents the overall functionality of a SAT engine using DCA. Simulation results on large industrial circuits are given in Section V.

II. SAT-BASED ATPG

SAT is a decision problem whose instance is a Boolean expression written using only AND,OR ,NOT ,variables

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

and parenthesis .To apply a SAT solver to a circuit-oriented problem, e.g., ATPG, the problem is formulated as a Boolean formula in CNF. A CNF formula φ in m Boolean variables is a conjunction of n clauses. Each clause is a disjunction of literals. A literal is a Boolean variable (x) or its complement ($\sim x$). The task of a SAT solver for a given _ is to find a satisfying assignment or to prove that no such assignment exists. The SAT instance φF test for test generation for fault φF consists of two parts: φF test = φc ・ φF. The CNF formula φc represents the functionality of the circuit, i.e., the correct copy according to [5]. In the following, the circuit-to-CNF transformation is briefly described. More information can be found in [5]. A Boolean variable is assigned to each connection in circuit C.

$$\varphi c = \Pi\, \varphi gi \,. \qquad (1)$$

The fault-specific constraints _F include the fault site as well as the faulty output cone. Improved SAT formulations for fault modelling were proposed in [6].
The CNF formula φF test is given to a SAT solver which determines whether there exists a solution, i.e., φF test is *satisfiable* (SAT), or prove that no solution exists, i.e., φF test is *unsatisfiable* (UNSAT). In case of SAT, the SAT solver provides a complete assignment from which the test can be extracted.

III. OVERALL FRAMEWORK FOR DYNAMIC CLAUSE ACTIVATION

A large part of the run time needed by a SAT-based algorithm stems from the CNF generation as reported in [6] and [7]. The transformation of the ATPG problem into a SAT problem is a significant overhead, in particular for easy-to detect faults. A very large number of SAT instances have to be built for each circuit for ATPG. Since the same parts of the circuit are considered very often, parts of the CNF formula can potentially be shared between faults. However, subsequent faults usually only share a small part or even no part in industrial practice due to the use of fault dropping. The DCA technique proposed in this article bypasses this problem by integrating large parts of the SAT instance generation into the search process. The CNF formula of the circuit is built only once and the necessary parts of the circuit are dynamically activated during the search process guided by structural information. The structural information is used to accelerate the search process as well.

The proposed SAT engine using DCA is divided into two parts: a circuit part and an algorithmic part. This is illustrated in Fig. 1. However, both parts are tightly integrated to allow for an efficient interaction. The circuit part contains the complete CNF formula of the circuit (φc) and serves as a database. The clauses of φc are kept in dedicated data structures providing structural information which is described in more detail in Section IV-A. No search algorithm operates on this CNF formula. The algorithmic part contains the search algorithm operating only on a local CNF formula φdyn representing the set of activated clauses. The search algorithm consists of the DLL algorithm [10] and modern SAT techniques such as conflict analysis and fast BCP and is only executed on φdyn. This is crucial to preserve the high efficiency of the search algorithm.

Each SAT solver can be taken as basis for the algorithmic part and extended to fit in a framework using DCA. As described above, the SAT instance φF test for test generation for fault F consists of two parts: the functionality. of C (φc) and the fault-specific constraints φF. Fault-specific constraints are those constraints which are added to the circuit CNF formula in order to generate a test for a specific fault:

$$\varphi F \text{ test} = \varphi c \text{ ・ } \varphi F \qquad (2)$$

In contrast to common SAT solvers, the proposed approach does not work on a static SAT instance φF test, but on dynamic SAT instance named φdyn. Initially, φdyn = φF holds for each fault.

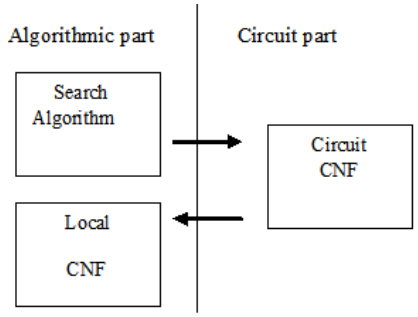

Fig.1 Division of the proposed SAT engine.

During the search process, φdyn is extended dynamically with clauses from φc using structural information. However, it is important to point out that once the clauses are activated; they are kept and not deactivated during the search (in contrast to [12]. This is described in more detail in Section IV. After solving the instance for one target fault, φdyn is cleared. All clauses are deactivated at once to avoid costly modifications of the SAT instance.

The DCA technique is integrated into the search process as follows. An activation request is sent to the circuit part whenever a variable is assigned. The circuit part answers with those clauses which have to be activated. By this, φdyn is dynamically extended during the search process. If a conflict in φdyn cannot be resolved, then the SAT instance is UNSAT, i.e., the fault is redundant, because any extension of φdyn would also result in UNSAT. By this, redundant faults can be classified much faster when the reason of the untestability is locally bounded. In contrast, SAT, is determined if no more activation requests are sent and all activated clauses are satisfied. This is similar to the concept of syntactic satisfiability proposed in [7]. The classical test generation flow as well as the alternated flow incorporating DCA is shown in Fig. 2(a) and (b), respectively. The time-consuming step of circuit-to-CNF transformation is done only once instead of for each fault. This step is replaced by the activation methodology which is crucial for the correct and efficient application of this technique. Only φF has to be extracted for each run. The size of the fault specific constraints φF is typically very small in relation to the size of the circuit CNF formula.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

IV. EFFICIENT ACTIVATION METHODOLOGY

In this section, the concrete activation methodology is described which is responsible for the dynamic extension of the SAT instance.

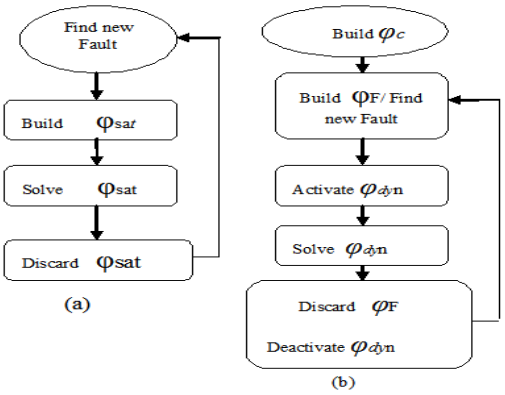

Fig. 2. SAT-based ATPG flow. (a) Classic. (b) Dynamic.

The methodology is influenced by the concept of justification. Justification corresponds to the search for an input assignment of a gate which results in the desired output value. If a value v is assigned to a variable x, those clauses should be activated which are needed to justify v. Since a variable is associated with a connection s and, consequently, s is the outgoing line of one gate φg, the set of clauses needed for justification is φg, i.e., the clauses of the preceding gate. Only the clauses of the preceding gates are activated. The clauses of the succeeding gates remain deactivated (if these clauses have not been activated before). This ensures that only clauses of the transitive fan-in cone become activated. Since the activation of clauses is done gate by gate, the CNF formula of a gate g is examined in more detail. This set of clauses can be divided into two parts

$$\varphi g = \varphi g0 \,.\, \varphi g1 \qquad (3)$$

The CNF formula $\varphi g0$ is satisfied if $g = 0$ and $\varphi g1$ is satisfied if $g = 1$. This is illustrated by the CNF formula for a NAND gate g in the following example. The application for other gate types is straightforward.

Example 1:

Consider the NAND gate $g = (a \cdot b)$. The following three clauses are needed to represent the functionality in CNF:

$$\begin{aligned} \omega1 &= (g + a + b) \\ \omega2 &= (g + a) \\ \omega3 &= (g + b) \end{aligned} \qquad (4)$$

For $g = 0$, $\omega1 = 1$ holds, i.e., $\omega1$ is satisfied, while $\omega2 = 1$ and $\omega3 = 1$ hold under the assignment $g = 1$. For $g = 0$, $\omega1 = 1$ holds, i.e., $\omega1$ is satisfied, while $\omega2 = 1$ and $\omega3 = 1$ hold under the assignment $g = 1$.

Therefore,

$$\begin{aligned} \varphi g0 &= \{\omega1\} \\ \varphi g1 &= \{\omega2, \omega3\}. \end{aligned} \qquad (5)$$

By this, only subsets of clauses of each gate have to be activated since some of the gate clauses are known to be satisfied. This correlates directly to the justification rules used in structural ATPG. For example, if the controlling value is assumed on the output of a primitive gate g, it is sufficient that only one input assumes the controlling value of g. The other inputs do not influence the value on the output and are don't care. The procedure is complete since the remaining clauses will be activated.IODCs (implicit observability don't care) is a great advantage for the search process since large parts of the search space can possibly be skipped. An example for the activation procedure with IODCs will be given after introducing the data structure for the efficient implementation of the proposed activation rules.

A. Structural Watch List

The structural watch list (SWL)—a new data structure which is part of the circuit part—is proposed for the use of the activation methodology. The implementation of the SWL is strongly influenced by the two-literal watch list used in modern SAT solver for fast BCP. Each entry in the SWL represents a list of clauses which should be activated when a specific activation request is sent, i.e., if a specific variable was assigned.

Therefore, each variable x in the circuit is associated with two entries in the SWL—each entry for =one literal (x and $\sim x$). The following example demonstrates the use of the SWL and the activation procedure.

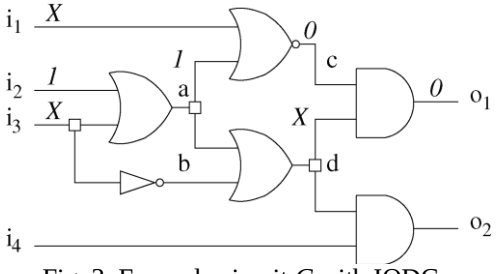

Fig. 3. Example circuit C with IODCs.

Example 2:

Fig.3 shows an example circuit. The corresponding SWL for this circuit is shown in Table I. Assume that the target is to justify the value 0 on output $o1$. According to the generated SWL, $\omega1$ is activated. In order to satisfy $\omega1$, variable c is assigned with 0 and $\omega4$ is activated.

TABLE 1
SWL ENTRIES FOR CIRCUIT C IN FIG. 3

$\omega1$	$o1\ (o1 + c + d)$	$\omega10$	$a\ (a + i2 + i3)$
$\omega2$	$o1\ (o1 + c)$	$\omega11$	$a\ (a + i2)$
$\omega3$	$o1\ (o1 + d)$	$\omega12$	$a\ (a + i3)$
$\omega4$	$c\ (c + i1 + a)$	$\omega13$	$b\ (b + i3)$
$\omega5$	$c\ (c + i1)$	$\omega14$	$b\ (b + i3)$
$\omega6$	$c\ (c + a)$	$\omega15$	$o2\ (o2 + d + i4)$
$\omega7$	$d\ (d + a + b)$	$\omega16$	$o2\ (o2 + d)$
$\omega8$	$d\ (d + a)$	$\omega17$	$o2\ (o2 + i4)$
$\omega9$	$d\ (d + b)$		

$i2 = 1$, $a = 1$, $c = 0$, $o1 = 0$.

The decision $a = 1$ is carried out. As a result, $\omega10$ is activated. The next decision is $i2 = 1$. Now, all activated clauses ($\omega1$, $\omega10$, $\omega4$) are satisfied and a solution is found, although not all variables in activated clauses are assigned.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 3, March 2014

The variables *i*1, *i*3, *d* are also contained in $\omega 1$, $\omega 10$, $\omega 4$ but they remain unconstrained and are consequently IODCs denoted by *X*. A solution can be found more quickly since parts of the CNF formula remain deactivated and large parts of the search space might be pruned by the use of IODCs.

V. SIMULATION RESULTS

The above circuit is simulated using verilog language. The proposed technique leads to a fast solving process for easy-to-detect faults without large transformation time as well as to the use of efficient SAT techniques to solve hard-to-detect faults.

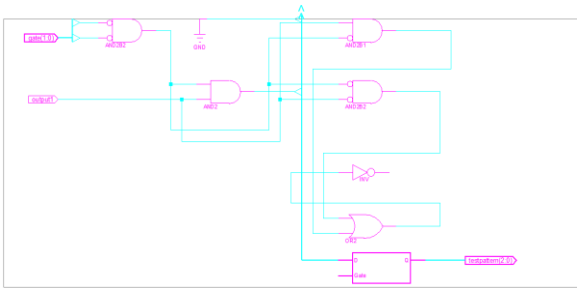

Fig.4 Schematic Diagram for circuit C

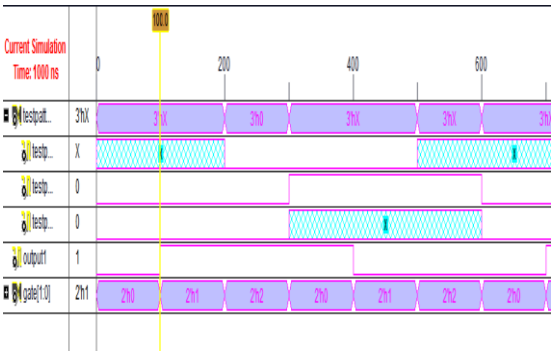

Fig.5 Generated Test Pattern for circuit C

TABLE 2
TESTING INPUTS FOR PARTICULAR OUTPUT

O1 (output)	i1	i2	i3	i4
	(inputs)			
1	1	1	0	X
0	X	X	1	1

The proposed approach is fast and provides very high fault efficiency and increased fault coverage even for the transition fault model. Additionally, a low number of specified bits is produced.

VI. CONCLUSION

SAT-based ATPG algorithms are very robust for hard-to test faults but suffer from the overhead for easy-to-test faults. Experimental results on large industrial circuits for stuck at faults as well as for transition faults have shown that the proposed techniques are able to significantly reduce the run time of SAT-based ATPG and reduce or even close the run time gap between structural and SAT-based ATPG approaches. At the same time, the number of unclassified faults is reduced to a minimum leading to very high fault efficiency and significantly increased fault coverage. This is especially important to satisfy the high quality demands of the industry and forms a basis for the application of SAT-based ATPG for new complex fault models.

REFERENCES

[1] Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," IEEE Trans. Comput., vol. 32, no. 12, pp. 1137–1144, Dec. 1983.
[2] M. H. Schulz, E. Trischler, and T. M. Sarfert, "SOCRATES: A
[3] highly efficient automatic test pattern generation system," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 7, no. 1, pp. 126–137, Jan. 1988.
[4] T. Larrabee, "Test pattern generation using Boolean satisfiability," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,
[5] Iouliia Skliarova and António B. Ferrari, Member, "A Software/Reconfigurable Hardware SAT Solver "2004, VOL. 12, NO. 4 ,pp. 502–518.
[6] R. Drechsler, S. Eggersgl¨uß, G. Fey, A. Glowatz, F. Hapke, J. Schloeffel, and D. Tille, "On acceleration of SAT-based ATPG for industrial designs," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 27, no. 7, pp. 1329–1333, Jul. 2008.
[7] S. Eggersgl¨uß, G. Fey, A. Glowatz, F. Hapke, J. Schloeffel, and R. Drechsler, "MONSOON: SAT-based ATPG for path delay faults using multiple-valued logics," J. Electron. Testing Theory Applicat., vol. 26, no. 3, pp. 307–322, 2010.
[8] A. Gupta, Z. Yang, and P. Ashar, "Dynamic detection and removal of inactive clauses in SAT with application in image computation," in Proc. Design Autom. Conf., 2001, pp. 536–541.
[9] S. Eggersgl¨uß, D. Tille, and R. rechsler, "Speeding up SAT-based ATPG using dynamic clause activation," in Proc. IEEE Asian Test Symp., Nov. 2009, pp. 177–182.
[10] S. Eggersgl¨uß and R. Drechsler, "Increasing robustness of SAT-based delay test generation using efficient dynamic learning techniques," in Proc. IEEE Eur. Test Symp., May 2009, pp. 81–86.
[11] S. Eggersgl¨uß and R. Drechsler, "Improving test pattern compactness in SAT-based ATPG," in Proc. IEEE Asian Test Symp., Oct. 2007, pp. 445–452.

BIOGRAPHY

V.M.Thoulath Begam received the B.E. degree in Electronics & Communication Engineering from BharadhiDasan University, Trichy in 2001 and the M.Tech. Degree in VLSI design from Sathyabama University, Chennai, in 2010, where she is currently working toward Ph.D. Degree. Her research interests include low power VLSI design, VLSI testing circuits, Fault coverage and BIST. She has published six papers in national conferences and two papers in international conferences.