SAT-based Automatic Test Pattern Generation

Rolf Drechsler, Stephan Eggersglüß, Görschwin Fey, and Daniel Tille

Institute of Computer Science
University of Bremen, 28359 Bremen, Germany
{drechsle,segg,fey,tille}@informatik.uni-bremen.de

Abstract. Due to the rapidly growing size of integrated circuits, there is a need for new algorithms for *Automatic Test Pattern Generation* (ATPG). While classical algorithms reach their limit, there have been recent advances in algorithms to solve *Boolean Satisfiability* (SAT). Because Boolean SAT solvers are working on *Conjunctive Normal Forms* (CNF), the problem has to be transformed. During transformation, relevant information about the problem might get lost and therefore is not available in the solving process.

In the following we briefly motivate the problem and provide the latest developments in the field. The technique was implemented and experimental results are presented. The approach was combined with the ATPG framework of NXP Semiconductors. Significant improvements in overall performance and robustness are demonstrated.

1 Introduction

Guaranteeing that a manufactured chip functions correctly is an important task. Therefore, every chip has to pass a post-production test during which a set of test patterns is applied to check for functional correctness. These test patterns are usually generated by algorithms for *Automatic Test Pattern Generation* (ATPG). Due to the ever increasing size of integrated circuits, the size of the problem instances which have to be handled by ATPG algorithms also increases. This results in growing run times of classical (structural) ATPG algorithms such as PODEM, FAN, SOCRATES and ATOM. As one alternative, ATPG based on *Boolean Satisfiability* (SAT) was introduced. The efficiency of SAT-based algorithms in the field of ATPG has recently been shown also for large industrial circuits. The efficiency of SAT-based ATPG is significantly driven by the development of powerful SAT engines in the last ten years. While the early approaches of SAT-based ATPG algorithms could only deal with Boolean logic, the more recent techniques handle tri-state elements and unknown values coming from the environment of a circuit based on a 4-valued logic.

SAT-based algorithms are not working directly on the circuit's structure, but on an equation in *Conjunctive Normal Form* (CNF). Therefore, the problem must be transformed into CNF. As a disadvantage, structural knowledge about the original problem which is used by classical algorithms to speed up the search process might get lost. On the other hand, due to the homogeneity of the representation as a formula, powerful algorithms can be applied to solve the problem. The inclusion of structural information leads to a faster solving process.

The main results of our work are briefly summarized in [1]. All the techniques required to create a powerful SAT-based ATPG tool for industrial needs are comprehensively discussed in [2]. The book also includes an introduction into ATPG and SAT.

Dagstuhl Seminar Proceedings 08351
Evolutionary Test Generation
http://drops.dagstuhl.de/opus/volltexte/2009/2015

Table 1. Experimental Results [1]

	SAT		FAN(de)		FAN(long)		FAN(de)+SAT	
Circuit	ab.	time	ab.	time	ab.	time	ab.	time
b14	0	0:19m	107	0:11m	7	1:42m	0	0:12m
b15	0	0:24m	619	0:11m	318	26:25m	0	0:18m
b17	0	2:22m	1382	1:41m	622	56:54m	0	1:58m
b18	0	22:30m	740	19:16m	270	41:40m	0	20:34m
b20	0	0:56m	225	0:35m	42	7:46m	0	0:44m
b21	0	0:59m	198	0:39m	43	6:48m	0	0:43m
b22	0	1:35m	284	1:07m	52	9:34m	0	1:14m
p44k	0	26:01m	12	4:58m	0	4:59m	0	5:55m
p49k	77	1:43h	3770	2:06h	162	2:38h	74	1:55h
p80k	0	9:43m	218	34:55m	21	39:13m	0	39:38m
p88k	0	9:33m	195	9:13m	38	12:40m	0	10:27m
p99k	0	6:50m	1398	6:02m	512	1:16h	0	7:25m
p177k	0	1:19h	270	16:06m	47	20:03m	0	19:03m
p462k	6	2:16h	1383	1:34h	423	2:07h	0	1:51h
p565k	0	2:23h	1391	2:21h	85	2:47h	0	2:47h
p1330k	1	5:05h	889	4:15h	144	4:28h	0	5:00h

Table 1 reports recent results for the standard ITC'99 benchmarks set (b*) and for industrial circuits (p*) that have been found to be hard cases for ATPG provided by NXP Semiconductors Germany GmbH. For the industrial circuits the number roughly corresponds to the number of gates in the circuit, e.g. p1330k has 1.3 million elements including gates, I/O, flip flops, buffers and tri-state elements. The SAT-based ATPG engine has been prototypically integrated into the ATPG framework of NXP Semiconductors that also includes a FAN-based ATPG engine. Compared are the four approaches: SAT-based engine (SAT), FAN-based engine with default parameters (FAN(de)), FAN-based engine with drastically increased resource limits (FAN(long)), and the combined approach combining both engines (FAN(de)+SAT). For each approach the number of faults that cannot be classified are reported in column (ab.=aborted) and the run time in CPU minutes (m) or hours (h), respectively.

The SAT-based is very effective and only leaves a few faults unclassified. In contrast FAN(de) aborted many faults and even increasing the resources does not help to classify all faults. But the run time is much smaller compared to the SAT-based approach. Finally, integrating FAN and SAT yields a robust ATPG framework. Almost all faults are classified at quite moderate run times. Even for the exception we were able to classify all faults by increasing the resource limits for the SAT-based approach.

Mandatory techniques on tuning the SAT-based approach for ATPG and on the integration are provided in [2].

References

1. Drechsler, R., Eggersglüß, S., Fey, G., Glowatz, A., Hapke, F., Schlöffel, J., Tille, D.: On acceleration of SAT-based ATPG for industrial designs. IEEE Transactions on Computer Aided Design of Circuits and Systems **27** (2008) 1329–1333
2. Drechsler, R., Eggersglüß, S., Fey, G., Tille, D.: Test Pattern Generation using Boolean Proof Engines. Springer (2009) to appear.