In IEEE European Test Workshop (ETW), pages 147-148, May 1998.

A Model and Algorithm for Computing Minimum-Size Test Patterns

Paulo F. Flores, Horácio C. Neto, Krishnendu Chakrabarty and João P. Marques Silva
Instituto Superior Técnico
Cadence European Laboratories/INESC
Lisbon, Portugal
{pff,hcn}@inesc.pt,kchakrab@hilsa.bu.edu,jpms@inesc.pt

Abstract —

This paper addresses the problem of test pattern generation for single stuck-at faults in combinational circuits, under the additional constraint that the number of specified primary input assignments is minimized. This problem has different applications in testing, including the identification of don't care conditions to be used in the synthesis of Built-In Self-Test (BIST) logic. The proposed solution is based on an integer linear programming (ILP) formulation which builds on an existing Propositional Satisfiability (SAT) model for test pattern generation. This ILP formulation is linear on the size of the original SAT model for test generation, which is linear on the size of the circuit. Nevertheless, the resulting ILP instances represent complex optimization problems, that require dedicated ILP algorithms. Preliminary results on benchmark circuits validate the practical applicability of the test pattern minimization model and associated ILP algorithm.

1. Introduction

Automatic test pattern generation (ATPG) for stuck-at faults in combinational circuits is now a mature field, with an impressive number of highly effective models and algorithms [5, 6, 9-12]. (A more comprehensive bibliographic review of recent ATPG algorithms can be found in [3].) Furthermore, besides being effective at detecting the target faults, recent ATPG tools have aimed the heuristic minimization (i.e. compaction) of the total number of test patterns required for detecting all faults in a circuit [2, 9, 10]. In general, the degree of test pattern compaction is expected to be related to the number of unspecified input assignments in each test pattern. Moreover, recent work on using deterministic test patterns for the synthesis of Built-In Self-Test (BIST) logic [2] also motivates the computation of test patterns for which the number of unspecified primary input assignments is minimized. Indeed, if the test set is used as input to a logic synthesis tool with the purpose of synthesizing BIST logic, then by maximizing the number of unspecified input assignments, i.e. by maximizing the don't care set of each test pattern, the logic synthesis tool is in general able to yield smaller synthesized logic. Thus the maximization of the don't care set of each test pattern, or conversely, the computation of test patterns of minimum-size, can have significant practical consequences.

Nevertheless, there exists no model or algorithm in the literature for computing test patterns for which the number of unspecified primary input assignments is maximized. Accordingly, the main objective of this paper is to propose a first attempt at solving this problem. We develop a new model for test pattern generation, based on propositional satisfiability (SAT), in the presence of unspecified input assignments. Afterwards, we derive an integer linear programming (ILP) model for maximizing the number of unspecified primary input assignments. Finally, we provide results for the IWLS'89 [7] and ISCAS'85 [1] benchmarks that justify using the proposed model in medium to large size combinational circuits. Besides its practical applicability, to our best knowledge this is the first formal non-heuristic model and algorithm towards computing minimum size test patterns.[1]

2. Model and Algorithm

In this section we briefly outline the integer optimization model for computing minimum-size test patterns. The main steps for constructing the model are as follows [3]:

1. The first step is to represent circuits and fault detection problems using Conjunctive Normal Form (CNF) formulas. In this paper, the model of [11] is assumed but the models of [5, 12] could also be used.
2. The next step is to develop a CNF model in which variables can have unspecified assignments. Notice that solving SAT requires that all variables must be specified. Consequently, a dedicated formal model was developed. (This model is detailed in [3].)
3. Afterwards, we apply the resulting CNF model to the representation of circuits and fault detection problems.
4. We can then map the resulting CNF into an ILP model. This step is straightforward, since clauses can always be viewed as algebraic inequalities.
5. Finally, we specify the cost function of the resulting integer optimization model so that the total number of specified assignments is minimized.

It can be shown that the proposed ILP model is indeed correct [3]. Furthermore, this model has a search space of $O\left(2^{2|P|}\right)$, hence significantly larger than the search space for plain fault detection. As a result dedicated ILP algorithms, targeted at highly constrained ILPs, have been developed [8]. These algorithms are built on top of Propositional Satisfiability (SAT) algorithms, that are specifically targeted at solving highly constrained instances of SAT.

1. This problem was addressed before by S. Hellebrand et al. in [4] but using a completely heuristic approach not based on a formal model.

Circuit	#PI	#F	ATALANTA			MTP				
			#R	#A	%X	#R	%X	D	% Op	sec/ fault
9symml	9	752	2	0	1.4	2	8.9	7.5	100	2.04
cht	47	820	0	0	93.6	0	94.2	0.6	98	1.00
cm138a	6	124	0	0	16.7	0	16.7	0.0	100	0.02
cm150a	21	232	0	0	68.4	0	71.0	2.6	100	1.55
cm163a	16	220	0	0	70.7	0	72.8	2.1	100	0.28
cmb	16	248	0	0	29.6	0	30.0	0.4	100	0.07
comp	32	480	1	0	24.0	1	39.6	15.6	2	10.64
comp16	35	960	0	0	30.7	0	32.9	2.2	4	13.66
cordic	23	342	0	0	30.7	0	40.2	9.5	37	6.28
cu	14	262	7	0	53.0	7	57.1	4.1	100	0.14
majority	5	54	0	0	8.5	0	8.5	0.0	100	0.01
misex1	8	224	0	0	49.8	0	54.4	4.6	100	0.17
misex2	25	422	0	0	73.5	0	75.8	2.3	100	0.20
mux	21	202	0	0	67.3	0	75.8	8.5	100	0.94
pcle	19	328	0	0	73.3	0	74.9	1.6	99	0.45
pcler8	27	400	0	0	78.1	0	79.2	1.1	98	1.97
c432	36	524	3	1	56.2	4	64.1	7.9	2	27.04
c499	41	758	8	0	17.1	8	19.5	2.4	0	33.71
c880	60	942	0	0	82.2	0	85.6	3.4	40	22.34
c1355	41	1574	8	0	13.3	8	15.2	1.9	0	64.86
c1908	33	1878	8	0	44.7	8	50.9	6.2	1	73.44
c2670	233	2746	97	20	92.0	117	93.0	1.0	25	83.46
c3540*	50	3425	134	0	74.6	134	77.3	2.7	15	16.81
c5315*	178	5350	59	0	92.6	59	92.9	0.3	14	9.34
c6288*	32	7744	34	387	22.2	34	22.5	0.3	1	37.22
c7552*	207	7550	77	181	86.9	131	89.4	2.5	4	17.48

Table 1: IWLS'89 and ISCAS'85 results

3. Experimental Results

The model described in the previous section has been integrated in a test pattern generation framework for the computation of minimum size test patterns referred to as *Minimum Test Pattern generator* (MTP), which uses the SAT-based ILP algorithm of [8] and the fault simulator provided with ATALANTA [6]. Table 1 contains the results for the IWLS'89 [7] and ISCAS'85 [1] benchmarks for both ATALANTA [13] and MTP. (In all cases MTP was run with a bound on the amount of search allowed. This allows MTP to identifying acceptable solutions, which in some cases are not optimal.) For each benchmark, all faults were targeted in order to allow for a meaningful comparison between the two algorithms. Columns #PI, #F, #R and #A denote, respectively, the number of primary inputs, faults, redundant faults and aborted faults. %X denotes the percentage of don't care bits; Δ denotes the variation in the don't care bit percentage from ATALANTA to MTP; %Op denotes the percentage of faults for which MTP was able to find the minimum-size test pattern. Finally, sec/fault denotes the average time spent by MTP on each fault. (Circuits marked with * indicate that MTP was run with a lower bound on the amount of search allowed, thus reducing the average CPU time per fault). Note that column #A is not included for MTP because no faults were aborted.

As can be observed, the proposed algorithm, MTP, allows the identification of minimum-size test patterns and hence it yields a significantly smaller number of specified input assignments. From the above results we can conclude the following:

- For some circuits the heuristics used by ATALANTA are extremely effective, and MTP can be used to formally show that.
- Whenever the main goal is maximizing the number of don't care bits, then MTP can be run on top of ATALANTA, thus in general allowing for further reductions in the number of specified bit assignments.

4. Conclusions

In this paper we introduce a new integer optimization model for computing minimum-size test patterns. Experimental results validate the practical applicability of the model. Additional research work includes developing more effective ILP algorithms and applying the proposed model and algorithm in BIST logic synthesis.

References

[1] F. Brglez and H. Fujiwara, "A Neutral List of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN," in *Proceedings of the International Symposium on Circuits and Systems*, 1985.

[2] K. Chakrabarty, B. T. Murray, J. Liu and M. Zhu, "Test Width Reduction for Built-In Self Testing", in *Proceedings of the International Test Conference*, October 1997.

[3] P. F. Flores, H. C. Neto and J. P. Marques Silva, "An Exact Solution to the Minimum-Size Test Pattern Problem," in *Proceedings of IEEE/ ACM International Workshop on Logic Synthesis*, June 1998.

[4] S. Hellebrand, B. Reeb, S. Tarnick and H.-J. Wunderlich, "Pattern Generation for a Deterministic BIST Scheme," in *Proceedings of the International Conference on Computer-Aided Design*, 1995.

[5] T. Larrabee, "Test Pattern Generation Using Boolean Satisfiability," *IEEE Transactions on Computer-Aided Design*, vol. 11, no. 1, pp. 4-15, January 1992.

[6] H. K. Lee and D. S. Ha, "On the Generation of Test Patterns for Combinational Circuits," Technical Report No. 12_93, Department t of Electrical Engineering, Virginia Polytechnic Institute and State University, 1993.

[7] IWLS'89 Benchmark Suite, available from http://www.cbl.ncsu.edu/ pub/Benchmark_dirs/LGSynth89/.

[8] V. Manquinho, P. Flores, J. P. M. Silva and A. Oliveira, "Prime Implicant Computation Using Satisfiability Algorithms," in *Proc. of the IEEE International Conference on Tools with Artificial Intelligence*, November 1997.

[9] I. Pomeranz, L. N. Reddy, S. M. Reddy, "COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits," in *IEEE Transactions on Computer-Aided Design*, vol. 12, no. 7, pp. 1040-1049, July 1993.

[10] M. H. Schulz and E. Auth, "Improved Deterministic Test Pattern Generation with Applications to Redundancy Identification," *IEEE Transactions on Computer-Aided Design*, vol. 8, no. 7, pp. 811-816, July 1989.

[11] J. P. M. Silva and K. A. Sakallah, "Robust Search Algorithms for Test Pattern Generation," in *Proceedings of the Fault-Tolerant Computing Symposium*, June 1997.

[12] P. R. Stephan, R. K. Brayton and A. L. Sangiovanni-Vincentelli, "Combinational Test Generation Using Satisfiability," *IEEE Transactions on Computer-Aided Design*, vol. 15, no. 9, pp. 1167-1176, September 1996.