In Proceedings of the IEEE Great Lakes Symposium on VLSI (GLS), pages 8-11, March 1999.

On Applying Set Covering Models to Test Set Compaction

Paulo F. Flores, Horácio C. Neto and João P. Marques-Silva

Instituto Superior Técnico
Cadence European Labs/INESC
1000 Lisboa, Portugal
e-mail: {pff,hcn,jpms}@inesc.pt

Abstract

Test set compaction is a fundamental problem in digital system testing. In recent years, many competitive solutions have been proposed, most of which based on heuristics approaches. This paper studies the application of set covering models to the compaction of test sets, which can be used with any heuristic test set compaction procedure. For this purpose, recent and highly effective set covering algorithms are used. Experimental evidence suggests that the size of computed test sets can often be reduced by using set covering models and algorithms. Moreover, a noteworthy empirical conclusion is that it may be preferable not to use fault simulation when the final objective is test set compaction.

1. Introduction

Test set compaction is a fundamental problem in digital system testing. In recent years, many competitive solutions have been proposed, most of which based on heuristic approaches [1, 5, 6, 8, 10, 11, 12, 13, 14, 15]. This paper studies the application of set covering models to the compaction of test sets, which can be used with any heuristic test set compaction procedure. For this purpose, recent and highly effective set covering algorithms are used. Experimental evidence suggests that the size of computed test sets can often be reduced by using set covering models and algorithms. Moreover, a noteworthy empirical conclusion is that it may be preferable not to use fault simulation when the final objective is test set compaction.

Let *C* be a combinational circuit and let *T* be a test set that detects all the single stuck-at non-redundant faults in *C*. After fabrication, the application of *T* to *C* ensures that the every single stuck-at non-redundant fault will be detected. In general the minimization of *T* is a key objective, since it reduces testing time and consequently overall testing costs. Approaches for minimizing *T* can be characterized by being based on exact models [5, 6, 10, 11, 12] or on heuristic approaches [8, 13, 14, 15]. Most practical solutions are based on heuristic approaches which are not guaranteed to compute a test set of minimum size, but which in practice perform extremely well [5, 13]. For heuristic approaches, and given a pre-computed test set *T* which is not known to be optimum, one can potentially remove redundant test patterns from *T*, thus obtaining a reduced test set *U*. Clearly *U* need not be the minimum-size test set for *C*. The existence of redundant tests in *T* is intrinsic to any fault simulation strategy used by the ATPG tool, since in general any fault simulation strategy is unable to guarantee the complete elimination of redundant test patterns. One approach for minimizing a pre-computed test set *T* has been proposed by D. Hochbaum in [6], and consists of casting the problem of removing redundant test patterns from a test set *T* as an instance of the set covering problem. Since the set covering problem can naturally be formulated as a 0-1 integer linear program, an integer programming approach based on linear programming relaxations was proposed and evaluated in [6]. Nevertheless, only very preliminary experiments were conducted. In particular, the effect of fault simulation on the ability of reducing the size of a test set was not evaluated. In this paper we review the set covering model for test set compaction. We then use a highly effective algorithm for the unate covering problem [4] and evaluate the application of the model in the simplification of test sets. Moreover, we study the relationship between the application of fault simulation and the ability of reducing the test set size. Experimental evidence, obtained on a large number of benchmark circuits, clearly indicates that the utilization of fault simulation in general reduces the ability for computing smaller test sets. This empirical result raises the question of which test-pattern generation strategy is best suited for computing highly compacted test sets.

The paper starts in the next section be describing a set covering model for test set compaction. Section 4 presents and analyzes experimental results of using the proposed set covering model. Section 5 concludes the paper.

2. Set Covering Model for Test Compaction

Let $F = \{f_1, \ldots, f_m\}$ be the set of stuck-at faults of a combinational circuit *C*, and let $T = \{t_1, \ldots, t_n\}$ be a pre-computed test set. Furthermore, let the faults detected by test pattern t_j be $F(t_j) = \{f_{j_1}, \ldots, f_{j_k}\}$. Consequently, the objective of the *test set compaction problem* is to find a a set of test patterns $U \subseteq T$, such that,

$$F = \bigcup_{t_j \in U} F(t_j) \quad (1)$$

and such that the size of U is minimum. This problem can naturally be mapped into an instance of the set covering problem. Indeed, define a matrix D where $d_{ij} = 1$ provided test pattern t_j detects fault f_i. Further, define a vector x of Boolean variables, with size $(1 \times n)$, such that $x_j = 1$ provided test pattern t_j is selected for inclusion in U. Consequently, our goal is to solve the following integer optimization problem,

$$\begin{aligned} &\text{minimize} && \sum x_j \\ &\text{subject to} && D \cdot x \geq 1 \\ & && x \in \{0, 1\}^n \end{aligned} \quad (2)$$

which can also be viewed as an instance of the set covering (or unate covering) problem. This model was first described in [6] and some results were obtained with a set covering algorithm based on linear programming relaxations. Nevertheless, other more efficient set covering algorithms can be used [4], which allow for larger test sets to be considered. This solution in turn enables considering different test pattern generation strategies which, in a preliminary phase, may generate a larger number of test patters that are later minimized with a set covering algorithm.

It should be noted that the solution of (2) does not necessarily yield the minimum size test set for the circuit, and in general it will not. In the case the test set contains *all* possible input patterns, then the solution of (2) is indeed the minimum size test set for the given circuit. Nevertheless, in practice the test set that is considered is provided by an ATPG tool, and hence denotes a small subset of the set of all the test patterns.

3. An Example

As an application example of the model proposed in the previous section, let us consider a combinational circuit with fault set $\{f_1, f_2, f_3, f_4, f_5\}$ such that the set of test patterns $\{t_1, t_2, t_3, t_4\}$ detects all faults. Let us assume further that the relation between test patterns and detected faults is as shown in Figure 1, where an entry (i, j) with value 1 indicates that test pattern t_j detects fault f_i, and an entry with value 0 indicates that the fault is not detected with t_j. Using this information, the resulting covering problem can be formulated as follows:

$$\text{minimize } \{x_1 + x_2 + x_3 + x_4\} \quad (3)$$

subject to the constraints,

	t_1	t_2	t_3	t_4
f_1	1	0	0	1
f_2	1	1	0	0
f_3	0	0	1	1
f_4	0	1	0	1
f_5	1	0	1	0

Figure 1: Covering table for a set of faults

Circuit	FS + No Comp		No FS + No Comp		No FS + Comp	
	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC
C432	78	60	520	41	184	44
C499	93	58	750	52	276	57
C880	69	50	942	—	116	44
C1355	131	93	1566	84	523	87
C1908	179	128	1870	—	520	116
C2670	159	111	2621	—	300	106
C3540	211	145	3291	—	137	126
C5315	1781	109	—	—	619	—
C6288	36	24	—	—	714	—
C7552	288	215	—	—	572	149

Table 1: Results for the ISCAS'85 circuits

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (4)$$

where each x_i, $1 \leq i \leq 4$, is a Boolean variable.

The minimum solution to this covering problem is $x_1 = 1$ and $x_4 = 1$, which indicates that t_1 and t_4 can be selected as a reduced set of test patterns for detecting *all* faults in the circuit.

4. Experimental Results

The model described in the previous section has been applied to test sets computed by Atalanta [9] under different operating conditions. Afterwards, a unate covering algorithm, Scherzo [4], has been applied to the different test sets, with the objective of minimizing those test sets. The experimental results of using Atalanta are shown in Table 1, in Table 2, and in Table 3, respectively for the ISCAS'85 [2], for the IWLS'89 [7] and for the ISCAS'89 [3] benchmark circuits. In these tables, FS indicates the utilization of fault simulation and COMP denotes

Circuit	FS + No Comp		No FS + No Comp		No FS + Comp	
	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC
9symml	93	78	750	75	153	75
alu4	132	104	2696	73	415	83
apex2	121	87	945	66	216	74
cht	21	14	820	—	14	11
cm138a	13	12	124	11	15	12
cm150a	42	33	232	33	48	36
cm163a	16	15	220	12	13	12
cmb	37	30	248	28	33	26
comp	62	56	479	48	75	54
comp16	93	72	960	51	115	74
cordic	52	43	342	42	60	41
cps	173	145	4640	133	264	140
cu	34	27	255	24	31	25
dalu	123	96	3740	—	193	86
duke2	128	101	1708	89	157	94
e64	70	68	1142	68	69	68
i2	213	208	760	208	215	208
majority	11	11	54	11	11	11
misex1	21	16	224	16	21	16
misex2	55	51	422	45	40	38
misex3	203	152	2583	134	449	141
mux	40	35	202	34	49	36
pcle	19	17	328	16	20	18
pcler8	25	19	400	18	21	19
seq	369	280	5908	239	549	263
spla	209	161	2522	143	325	151
term1	77	54	702	33	71	43
too_large	145	103	1117	85	233	95
unreg	17	14	448	—	12	10

Table 2: Results for the IWLS'89 circuits

the application of test compaction using simple dominance relations. NO FS indicates that fault simulation is not applied, and that all faults are targeted. (Note that ATALANTA [9] computes test patterns with don't cares, which enhance dominance between test patterns. Moreover, dominance-based test compaction is useless whenever fault simulation is also applied.) For each experiment the total number of test patterns (#T) is shown, either obtained by the ATPG tool, or after applying the test set compaction tool (MTSC) that runs Scherzo. The CPU time allowed for Scherzo was 4,000 seconds on a SUN Ultra I/ 170 workstation. Table entries with '—' indicate that the CPU time was exceeded.

As can be concluded from the tables of results, test set compaction can in general yield significant savings in the number of test patterns, even when fault simulation is

Circuit	FS + No Comp		No FS + No Comp		No FS + Comp	
	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC	#T w/ ATPG	#T w/ MTSC
S208	45	34	217	34	59	35
S298	33	28	308	25	38	29
S344	27	24	324	14	31	22
S349	26	23	330	15	31	21
S382	44	34	399	26	42	30
S386	82	68	384	65	92	68
S400	45	36	418	24	43	30
S420	92	73	455	70	125	72
S444	48	32	460	24	42	32
S510	69	61	564	56	84	58
S526	85	62	554	51	81	62
S526n	80	59	553	50	79	58
S641	76	58	465	43	53	43
S713	78	58	543	42	60	40
S820	177	111	850	96	143	105
S832	168	113	856	102	147	106
S838	168	153	931	148	252	145
S953	104	87	1079	76	166	82
S1196	189	150	1242	129	251	143
S1238	197	155	1286	136	255	149
S1423	88	66	1501	—	110	58
S1488	189	125	1486	105	162	112
S1494	172	120	1494	102	165	110
S5378	316	246	4563	—	287	178
S9234	497	374	6475	—	422	241

Table 3: Results for the ISCAS'89 circuits

applied. Moreover, by not applying fault simulation, and thus by having a significantly larger initial test set, the test set compaction procedure is able, in the vast majority of cases, to compute test sets smaller than those obtained with fault simulation. Nevertheless, for the larger circuits, the non-utilization of fault simulation yields a very large number of test patterns, which the set covering algorithm may then be unable to simplify. One solution to overcome this problem is to compact test patterns by using dominance relations (columns NO FS + COMP in the tables). In this case, the set covering algorithm is able to optimally solve a larger number of problem instances. Nevertheless, the number of test patterns may still be too large for the set covering algorithm to handle. One additional simplification technique is to partition the test set into k subsets and simplifying each subset separately. Afterwards, the process is repeated for pairs of subsets. The process is repeated until one reduced test set is obtained.

As one final remark, we should note that the improvements obtained with the test set compaction procedure also result from the ATPG tool computing test sets that are

significantly larger than the optimum. Nevertheless, the set covering procedure of Section 2 can be applied to test sets computed by any ATPG tool (e.g. COMPACTEST [13] or MinTest [5]) whenever these test sets are known not to be optimum or whenever fault simulation is not applied. We note, however, that for any of these ATPG tools one should have the ability to conditional apply fault simulation.

5. Conclusions

This paper describes and evaluates a set covering model for test set compaction. The model can be used by a post-processing tool, for further compacting test sets obtained with ATPG algorithms. Experimental evidence indicates that in general additional test set compaction can be achieved. Moreover, by augmenting the size of the initial test set, e.g. by not using fault simulation and targeting all faults, we were able to compute test sets that are in general smaller than those obtained when fault simulation is used. Hence, whenever the main objective is test set compaction one may consider utilizing highly efficient ATPG algorithms, targeting all faults (i.e. no fault simulation) and applying a set covering algorithm for test set minimization. Despite this interesting result, set covering is an NP-hard problem and consequently existing algorithms may be unable to handle large test sets. For this problem the paper describes different techniques, including precompaction of test sets based on dominance relations and test set partitioning. Future work will involve the evaluation of other publicly available test pattern generators, that specifically target test set compaction, with the goal of studying the relationship between heuristic procedures and formal models and algorithms for the test set compaction problem.

References

[1] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.

[2] 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(ISCAS)*, 1985. (ISCAS'85 Benchmark Suite available from http://www.cbl.ncsu.edu/pub/Benchmark_dirs/.)

[3] F. Brglez, D. Bryan and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits", in *Proceedings of the International Symposium on Circuits and Systems*, pp. 1929-1934, May 1989. (ISCAS'89 Benchmark Suite available from http://www.cbl.ncsu.edu/pub/Benchmark_dirs/.)

[4] O. Coudert, "On Solving Covering Problems," in *Proceedings of the Design Automation Conference*, June 1996.

[5] I. Hamzaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," *in Proceedings of the International Conference on Computer-Aided Design*, November 1998.

[6] D. Hochbaum, "An Optimal Test Compression Procedure for Combinational Circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 10, pp. 1294 - 1299, October 1996.

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

[8] B. Krishnamurthy and S.B. Akers, "On the Complexity of Estimating the Size of a Test Set," in *IEEE Transactions on Computers*, vol. 33, no. 8, pp. 750-753, August 1984.

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

[10] Y. Matsunaga, "MINT—An Exact Algorithm for Finding Minimum Test Set," in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E76-A, no. 10, pp. 1652-1658, October 1993.

[11] J. P. Marques-Silva, "Integer Programming Models for Optimization Problems in Test Generation," in *Proceedings of the IEEE Asian-South Pacific Design Automation Conference*, pp. 481-487, February 1998.

[12] I. Pomeranz and Z. Kohavi, "The Minimum Test Set Problem for Circuits with Nonreconvergent Fanout," in *Journal of Electronic Testing: Theory and Applications*, vol. 2, pp. 339-349, 1991.

[13] 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.

[14] 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.

[15] G.-J. Tromp, "Minimal Test Sets for Combinational Circuits," in *Proceedings of the International Test Conference*, pp. 204-209, 1991.