In IEEE European Test Workshop (ETW), pages 34-36, May 1998.

Exploiting Don’t Cares in Test Patterns to Reduce Power During BIST

José C. Costa, Paulo F. Flores, Horácio C. Neto, José C. Monteiro and João P. Marques Silva
Instituto Superior Técnico
Cadence European Laboratories/INESC
Lisbon, Portugal
{jccc,pff,hcn,jcm,jpms}@algos.inesc.pt

Abstract —

For a significant number of electronic systems used in safety-critical applications circuit testing is performed periodically. For these systems, power dissipation due to Built-In Self Test (BIST) can represent a significant percentage of the overall power dissipation. One possible solution to address this problem consists of test pattern reordering with the purpose of reducing the amount of power dissipated during circuit testing. By reordering test patterns one is able to find test sequences for which power dissipation is minimized. Moreover, a key observation is that test patterns are in general expected to exhibit don’t cares, which can naturally be exploited during test pattern reordering. In this paper we describe efficient algorithms for test pattern reordering in the presence of don’t cares. Preliminary experimental results amply confirm that the power savings due to test pattern reordering using don’t cares can be significant.

1. Introduction

This paper addresses the problem of reducing power dissipation during Built-In Self Test (BIST). In applications where BIST is applied periodically, BIST power reduction techniques can significantly contribute to overall power reduction. It is well-known that power dissipation during circuit testing can be achieved by reordering the testing sequence [2]. However, existing approaches explicitly assume completely specified test patterns. In general, however, test patterns need not be completely specified. Indeed, incompletely specified test patterns can play a significant role in BIST design [1]. The purpose of this paper is to describe techniques for BIST power reduction in the presence of don’t cares. Experimental evidence obtained on a significant number of benchmark circuits clearly shows that exploiting don’t cares in test patterns can lead to very significant savings in overall power dissipation.

2. Model and Algorithms

Let $\{T_1, T_2, \ldots, T_m\}$ be a given sequence of *completely specified* test patterns. The problem of power reduction during BIST can be formulated as the identification of a permutation $\langle i_1, \ldots, i_m \rangle$ such that the power of applying a sequence of test patterns $\langle T_{i_1}, T_{i_2}, \ldots, T_{i_m}, T_{i_1} \rangle$ is minimal over all possible permutations. In the model we derive below we explicitly assume that power dissipation is proportional to the Hamming distance between test patterns. Even though this assumption does not necessarily hold true in general, it is amply confirmed by the experimental results given in Section 3.

The problem of power reduction can naturally be reduced to the traveling salesman problem (TSP). Indeed, by viewing each test pattern as a vertex in a graph, and the Hamming distance between each pair of test patterns as the weight between those two vertices in the graph, then the sequence that minimizes power dissipation during BIST corresponds to a tour of least weight in the graph, i.e. the solution to the TSP. Consequently, different polynomial-time approximation algorithms can be used [4].

This simple model explicitly assumes that test patterns are completely specified. Hence, if test patterns have unspecified bits, then the straightforward reduction to the TSP no longer holds true. Indeed, edge weights in the graph representation of the test pattern reordering problem can now be viewed as conditional values which depend on the final value of bits unspecified in each test pattern. To solve this new optimization problem we propose the following heuristic procedure which is based on TSP approximation schemes:

1. Use a dedicated algorithm for computing a test set where each test pattern contains don’t cares [3].
2. Apply a heuristic procedure for identifying an initial tour. Several different heuristics are described below.

Circuit	Completely specified (ordered vs. unordered)		Incompletely specified versus ordered completely specified					
	#V	%PS	#V	% PS				
				H1	H2	H3	H4	H5
9symml	78	48.1	79	3.7	5.1	12.5	5.8	9.9
cht	17	10.4	10	16.5	19.0	21.0	9.2	10.1
cm138a	12	18.6	12	11.9	26.7	19.0	7.4	32.3
cm150a	34	10.7	38	36.6	44.7	45.6	47.8	48.8
cm163a	14	9.2	14	39.7	43.2	43.8	42.6	44.3
cmb	29	43.4	26	14.7	16.8	21.0	16.8	14.4
comp	57	28.6	60	56.5	53.7	53.7	53.4	52.3
comp16	71	36.5	99	38.4	40.4	40.7	40.6	41.9
cordic	42	35.4	46	50.3	53.2	56.9	51.2	52.9
cu	27	37.3	26	21.2	30.3	35.8	42.4	37.6
majority	11	14.2	11	21.5	20.6	21.5	17.5	25.0
misex1	16	17.8	17	23.0	12.4	29.6	19.0	23.9
misex2	49	26.8	39	42.3	48.2	54.2	53.5	49.9
mux	35	18.4	37	32.7	36.8	36.8	38.5	27.4
pcle	19	23.1	19	22.8	33.8	31.8	32.1	32.7
pcler8	19	8.4	23	35.8	37.4	38.5	41.0	49.0

Table 1: ATALANTA results for the MCNC benchmarks

Circuit	Completely specified (ordered vs. unordered)		Incompletely specified versus ordered completely specified					
	#V	%PS	# V	% PS				
				H1	H2	H3	H4	H5
9symml	78	48.1	92	0.3	1.3	7.7	0.5	10.5
cht	17	10.4	11	17.4	28.3	15.0	18.9	22.0
cm138a	12	18.6	12	18.5	23.6	32.3	19.6	25.7
cm150a	34	10.7	36	52.9	49.5	53.3	49.5	53.3
cm163a	14	9.2	13	50.8	56.5	50.8	50.8	58.5
cmb	29	43.4	28	48.5	44.4	37.5	43.6	31.1
comp	57	28.6	72	45.4	41.4	43.4	42.7	41.1
comp16	71	36.5	108	37.7	35.4	42.1	39.8	37.2
cordic	42	35.4	55	58.3	60.0	54.6	61.4	58.0
cu	27	37.3	30	41.3	25.9	41.2	37.0	39.2
majority	11	14.2	11	18.6	14.2	22.1	10.4	23.7
misex1	16	17.8	19	32.5	33.4	29.8	30.1	32.6
misex2	49	26.8	40	51.2	53.1	54.0	48.4	56.0
mux	35	18.4	36	37.0	43.4	36.4	38.4	26.1
pcle	19	23.1	19	32.0	-4.0	-4.0	-4.0	16.9
pcler8	19	8.4	24	23.2	22.2	31.6	22.2	23.2

Table 2: MTP results for the MCNC benchmarks

3. For the initial tour, compute the tour cost by specifying don’t care bits which minimize the distance between consecutive test patterns.
4. Use the 2-opt local-search approximation algorithm for the TSP [4] to reorder the test patterns.
5. At each step of the 2-opt algorithm, and while the total tour cost is being reduced, repeat step 3.

The following initial ordering heuristics have been implemented:

1. Random ordering.
2. Decreasing order of don’t cares in each test pattern.
3. Apply heuristic 2. Afterwards, greedily select the next test pattern as one that minimizes the distance from the current test pattern.
4. For each bit position, set don’t care bits to the bit that occurs more often. Afterwards order the test vectors approximating Gray coding.
5. Same as heuristic 4, but without ordering the test vectors. Afterwards, the Christofides TSP approximation algorithm [4] is used for defining an initial tour.

3. Experimental Results

The results of applying the different variations of the algorithm to the MCNC [6] benchmark circuits are shown in Table 1 and Table 2. Table 1 contains the power savings[1] results from using ATALANTA [5] to generate the incompletely specified test patterns, whereas Table 2 contains the results from using MTP [3] to generate test patterns with the maximum number of don’t cares.

The columns labeled **completely specified** indicate the percentage power savings (**%PS**) that result from ordering a sequence of completely specified test patterns, and the number of computed test vectors (**#V**). For this experiment the Christofides approximation algorithm [4] was used. The columns labeled **incompletely specified** indicate the power savings from exploiting the don’t cares in incompletely specified test patterns over an already ordered sequence of completely specified test patterns. For this experiment the algorithm described in previous section was used, and the

1. Note that these results correspond to the actual power dissipation obtained from simulated test sequences.

Circuit	Completely specified (ordered vs. unordered)		Incompletely specified versus ordered completely specified					
	#V	%PS	#V	% PS				
				H1	H2	H3	H4	H5
c432	58	16.9	75	36.7	48.6	43.5	41.3	43.0
c499	60	28.9	61	18.3	14.8	17.4	18.5	15.9
c880	51	10.4	79	52.1	46.8	44.9	44.9	54.2
c1355	94	30.5	96	9.7	7.8	10.8	11.4	10.7
c1908	128	27.4	175	44.8	45.2	50.3	44.7	50.8
c2670	117	14.2	156	68.0	68.5	68.7	66.2	69.4
c3540	159	18.7	253	30.7	33.6	44.2	33.9	47.4
c5315	116	11.0	158	50.1	49.9	52.9	50.7	53.6
c6288	25	18.6	54	55.6	57.7	57.5	57.1	53.9

Table 3: ATALANTA results for the ISCAS85 benchmarks

different initial ordering heuristics (**H1** through **H5**) were considered.

As can be readily concluded, large power savings ranging from 30% to 60% are achieved in most cases. This is particularly significant since these results measure the percentage power savings over the *already ordered* sequence of test patterns.

Moreover, from the two tables of results, we can also conclude that MTP [3] in general permits greater power savings than ATALANTA [5]. This should be expected since MTP necessarily computes test patterns with a larger number of don’t cares.

The power savings results for ISCAS’85 [7] benchmark circuits are shown Table 3, where ATALANTA is used to generate the test patterns, with and without don’t cares. From this table we can conclude that for the majority benchmarks the power savings are between 40% and 60%. Consequently, we can conclude that test pattern reordering in the presence of don’t cares leads to large power savings over already ordered test sequences

Furthermore, we noticed that the percentage power savings in general increases as the size of the circuit and number of test patterns increases. Hence, for large circuits we expect the proposed power reduction algorithm to lead to similar or greater power savings. Regarding the heuristics proposed in Section 2 for constructing the initial tour, the results do not identify a clear best heuristic, even though the greedy heuristic **H3** performs better in most cases. Finally, these experimental results clearly indicate that exploiting don’t cares in sequences of test patterns may prove extremely useful whenever power reduction is the main objective.

4. Conclusions

In this paper we propose a model and algorithm for test pattern reordering in the presence of don’t cares. An immediate application is the reduction of dissipated power during BIST. Even though we describe a simple heuristic approach for solving this problem, preliminary experimental results indicate that very significant savings in dissipated power can be achieved by applying the proposed procedure.

Additional research work entails the introduction of a more detailed model, intended to accurately predict power dissipated during transitions between test patterns, and studying alternative algorithmic solutions.

References

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

[2] S. Chakravarty and V. Dabholkar, “Minimizing Power Dissipation in Scan Circuits During Test Application,” in *International Workshop on Low-Power Design*, April 1994.

[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] D. S. Johnson and L. A. McGeoch, “The Traveling Salesman Problem: A Case Study in Local Optimization,” in *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra (eds.), John Wiley and Sons, 1996.

[5] 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.

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

[7] 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/ISCAS85/.)