An evolutionary approach to the design of on-chip pseudorandom test pattern generators

Michele Favalli
DI – University of Ferrara
Via Saragat, 1 – 44100 Ferrara, Italy

Marcello Dalpasso
DEI – University of Padova
Via Gradenigo, 6A – 35131 Padova, Italy

Weighted pseudorandom test generation (WPRTG) uses test sequences characterized by non-uniform distributions of test vectors in order to increase the detection probability of random resistant faults. Such non-uniform distributions are characterized by the values of signal probability of the CUT inputs (weights). Since different faults may require different distributions, a (small) number of distributions is typically used [1]. The weights of such distributions are identified by analyzing the CUT. The corresponding pseudorandom sequences are typically obtained by inserting a combinational network between the TPG and the CUT.

Several different methodologies have been proposed in order to calculate the weights. Some approaches make use of deterministic test sequences [2]. Another class of heuristics, instead, makes use of numerical optimization strategies to determine the set(s) of weights [1]. More recently, genetic algorithms have been identified to provide a good solution to weights selection [3]. All such methods evaluate only the first order coefficients of the distribution(s) and may suffer from a few problems. In particular, the detection of some random resistant faults may strongly depend on signal correlations. Even if the effects of signal correlations can be reduced, some problems are still in order. Consider, for instance, a fault that can be detected by a test vector and its complement. Any WPRTG method using signal probability evaluation would provide (when targeting such a fault) the same coefficients of a uniform distribution.

Conversely, the actual correlations of the TPG output signals may be successfully exploited by the WGU if it is properly designed. In fact, this is a good way to catch such faults whose detection is mainly dominated by second order coefficients such as in the above mentioned case.

To this purpose, instead of optimizing weight sets, we directly optimize the actual test sequence by using an evolutionary approach where we make the WGU (and consequently the actual test sequence) to evolve in order to achieve the best test efficiency. As a secondary evolution target, we also consider the minimization of the cost of the WGU.

Differently from the genetic-based approaches [3], where only numerical coefficients are computed, we have used an evolutionary programming (EP) algorithm that directly evolves the WGU network. In fact, evolutionary approaches have been shown to be effective in the design of digital circuits. In particular, we evolve a population were each individual represents a possible WGU and the fitness function considers the fault coverage as a primary target and the test length and the cost of the WGU as secondary ones. The fault coverage is here evaluated by means of fault simulation.

The proposed approach has been applied to the ISCAS combinational benchmarks set. Table 1 shows, for each of the considered benchmarks, the fault coverage (as evaluated on detectable faults only) (C), the number of generations (Γ), the gate count (G) and the test length (L) of the best individual. The last column of the table shows the best test length in the literature for each benchmark (OBL).

Bench	C (%)	Γ	G	L	OBL
c432	100.00	643	37	73	352
c499	100.00	278	109	158	768
c880	100.00	538	147	124	512
c1355	100.00	295	143	490	960
c1908	100.00	77	315	764	3296
c2670	99.70	349	1167	27904	5504
c3540	100.00	587	68	788	2400
c5315	100.00	257	415	604	2080
c6288	100.00	40	83	28	39
c7552	98.76	30	666	9187	2110

Tab. 1 Best individual (WGU) obtained from genetic programming. The others' best length reports the best results in the literature.

The results show the feasibility of the proposed approach. In the c2670 and c7552 cases, the results are worst. This is essentially a computational problem: these circuits, in fact, contain faults that are more random resistant than the other benchmarks. Therefore, a larger number of generations is required to achieve an optimum. In order to solve this problem, we are considering two possible strategies in order to bias the evolution process with some information regarding the undetected faults. The former makes use of deterministic test generation to help the evolution process. The latter make use of dynamic testability measures regarding undetected faults, thus merging an evolutionary ATPG process with the WGU evolution.

References

[1] H.-J. Wunderlich, "Multiple Distributions for Biased Random Test Patterns," in *Proc. of Int. Test Conf.*, pp. 236 – 244, 1988.

[2] F. Brglez, C. Gloster, and G. Kedem, "Built-In Self-Test with weighted random pattern hardware," in *Proc. of IEEE Int. Conf. On Computer Design*, pp. 161 – 166, 1990.

[3] C. Okmen, M. Keim, R. Krieger, and B. Becker, "On optimizing BIST-architecture by using OBDD-based approaches and genetic algorithms," in *Proc. of IEEE VLSI Test Symposium*, pp. 426–430, 1997.