The Use of VHDL Models for Design Verification[1]

Balázs BENYÓ
Department of Informatics
Széchenyi College
Hédervári út 3., 9026 Győr
Hungary
benyo@mit.bme.hu

József SZIRAY
Department of Informatics
Széchenyi College
Hédervári út 3., 9026 Győr
Hungary
sziray@szif.hu

Abstract – The correct and consistent design of computer hardware in a safety-critical system is extremely important. This feature necessitates a profound verification process related to the logic design of such a hardware. So far much work has been done on various verification methods, both formal and informal. This paper is intended to present an informal approach which can be used in the design process of digital networks.

Our verification method relies on automatic test-pattern generation (ATPG), where the VHDL model of the actual digital circuit is taken into consideration.

The test patterns of the circuit are generated in the earliest phase of the design process. The verification is done by means of test runs that are performed on the VHDL model. In a test run the circuit description to be verified is compared with a model verified in a previous test run, or with the original specification of the circuit. The VHDL models are simulated by a VHDL simulator and the tester compares the output of the simulated models.

The quality of the verification highly depends on the selected ATPG algorithm. In our paper two approaches are presented: a random ATPG algorithm, and a deterministic one. The deterministic algorithm is based on the traversal of the Control Flow Graph of the circuit description. The so-called Path Testing algorithm is used which was used originally for software test generation.

The paper presents both the verification and the test pattern generation methods. The results are illustrated by real examples.

I. DESIGN VERIFICATION

Design verification is the process of proving the correctness of the digital system design, that is, proving the compliance of a former specification with the design results. There are two alternative ways for doing it:

- formal design verification [1], and
- informal design verification.

In this paper we will deal with informal design verification.

The informal design verification testing requires a set of input vectors, the so-called test vector set, to prove the correctness of the circuit. There are two alternative ways to generate the test vector set:

- deterministic test vector generation, and
- random test vector generation.

Our research work focused on both the random and deterministic test vector generation based design verification.

II. RESULTS

A. Random Test Generation

The achieved results show that random test generation is a viable alternative technique in the field of design verification. The behavior of the random method in the design verification testing is very similar to the behavior of the methods in hardware testing.

The newly-developed technique for the estimation of the minimal size test set resulting in proper fault coverage with the given confidence level has also proved to be applicable in practice. By using this estimation technique, the quality of the generated test set can be approximately defined without the time consuming fault coverage evaluation.

A simulation-based fault coverage evaluation method has been introduced for the given high-level fault model, allowing the optimization of the random test set. Our experiences show that the time requirement of random test generation and test set optimization is not considerably higher than the time of deterministic test generation.

On the other hand, in the case of more sophisticated high-level fault models, the unfeasibility or high-complexity of the fault coverage evaluation may inhibit the optimization of the fault coverage.

In other cases, as an optimal solution, the combination of the random and deterministic test generation is suggested. The reason for it is that random test generation offers a similarly cost-effective alternative to algorithmic test generation at high level as at the gate level. In this setup the random test generator would provide test vectors until the efficiency of the process is decreased to a user-defined value. After that the deterministic algorithm would generate test vectors for covering the remaining faults.

B. DeterministicTest Generation

A new high-level constraint based test generation algorithm has been developed which can be an alternative way for design verification testing.

Through Direct Component Testing, i.e., combining CFG path predicate constraint sets with constraint sets describing the effective test set of the components, a new possibility of using structural information for the design verification test generation was introduced. The described algorithm is an efficient solution of information compaction where the structure of the designed circuit is involved.

Experimental results show that the test sets generated by the new deterministic algorithm yield a sufficiently high degree of gate-level fault coverage.

The developed special benchmarks parameterized by the bit width of the circuits served as a perfect test bed for estimating the dependency of the computation complexity of the test generation on the size of the circuit.

III. FURTHER DETAILES

[1] B. Benyó: Test Pattern Generation Based On High Level Hardware Description, CsC (PhD) Thesis, Budapest, 1997 (www.mit.bme.hu/~benyo)

[1] This work has been supported by the grant F-029739 of the National Scientific Research Foundation (OTKA) and by the Bolyai Foundation. The authors also wish to thank Dr. András Pataricza for his valuable advice in doing this research work.