Practical Works for On-Line Teaching Design and Test of Digital Circuits

A. Jutman, R. Ubar

Tallinn Technical University
ESTONIA

V. Hahanov, O Skvortsova

Kharkov National University of
Radio Electronics, UKRAINE

ABSTRACT

A conception of practical works for teaching design and test of digital circuits is presented. The works cover essential topics in testing and diagnostics field. They are meant for improving the skills of students to be educated for hardware and SoC design in test related topics. Six practical works are described. The works itself are based on two diagnostic software packages that were developed at Tallinn Technical University and Kharkov National Technical University of Radio Electronics. A brief description of these packages is given. All the training materials can be accessed over Internet and therefore can be used by students at any time and any place. The free-access basis and self-contained nature makes it much easier to implement this course in foreign universities with minimum help from our side.

1. INTRODUCTION

Rapid advances in areas of deep-submicron electronic technology and design automation tools enabled engineers to design larger, more complex integrated circuits. Until recently, most electronic systems consisted of one or multiple printed circuit boards, containing multiple integrated circuits (IC) each. Recent advances in IC design methods and technologies allow to integrate these complex systems onto one single IC. These developments are driving engineers toward new System on a Chip (SOC) design methodologies. SOC is seen as a major new technology and the future direction for the semiconductor industry. Within the next several years, SOC designers will cut new product development cycle time from an average of 10 months today, to just four months by 2004. The key to this forecast becoming a reality is in placing the power in the hands of a SOC designer.

On the other hand, the more complex are getting electronic systems the more important become problems of test and design for testability, as costs of verification and testing are getting the major component of design and manufacturing costs of a new product. Today, design and test are no longer separate issues. The emphasis on the quality of shipped products, coupled with the growing complexity of system design, require testing issues to be considered early in the design process.

At present, most VLSI and system designers know little about testing, so that companies frequently hire test experts to advise their designers on test problems, and they even pay a higher salary to the test experts than to their VLSI designers [1]. This reflects the today's university education: everyone learns about design, but only truly dedicated students learn test. Entering into the SOC era means that test must now become an integral part of the VLSI and system design courses. The next generation of engineers involved with VLSI technology should be made aware of the importance of test. They must be specially trained in test technology to enable them to produce high quality, defect-free products.

The National Science Foundation in USA held a workshop in 1998 where it was stated that the present level of "test coverage" in the computer engineering education in USA was inadequate. As a consequence to this statement, a special panel was organized at the International Test Conference (ITC) in 1999 how to enhance the coverage of test related topics in computer engineering education [2].

In the following we present a conception of internet-based „Home Laboratory": a downloadable set of diagnostic tools and HTML-based practical work descriptions. The laboratory course is designed to improve the skills of students to be educated for hardware and SOC design in test related topics.

The next section describes the idea of the overall concept of the training system. Short descriptions of each practical work from the package is given in Section 3. Section 4 contains information on PC-based software packages we use in our course. A couple of examples of work with PC-based tools are given in Section 5. Section 6 brings concluding remarks.

2. THE CONCEPT

The laboratory training course is meant to help students to obtain hands-on experience in basics of testing and diagnostics, design for test, and designing embedded self-test architectures. This course must be an essential supplement to the theoretical knowledge students receive from lectures. All the designed training manuals can be accessed over Internet [7] and therefore can be used by students at any time and any place. The required tools can be preinstalled in universities while at home they can be downloaded for free. The free-access basis and self-contained nature makes it much easier to implement this course also in foreign universities with minimum help from our side.

All laboratory training materials are composed in HTML language providing links to related topics in theoretical notes, which are a complementary part of the lab training. Therefore students can reach the exact place in the theory instantly refreshing the theoretical knowledge for a particular part of the work.

The manuals contain figures and tables to visualize the content of the works. Comprehensive examples and detailed descriptions are also very helpful for students and allow them to work fully independently of the teacher. Each software component, which is required to perform certain operations in the workflow is described extensively in help sections.

A well-structured layout of the training manuals is clear and informative.

- *Objectives* – this part explains which skills and practical knowledge a student can acquire during this work.
- *Introduction* – gives the basic information about the subject of the work, depicts industrial arias where it is used, etc.
- *Work description* – introduces the general flow of the work along with tools to be used and circuits to be examined.
- *Steps* – this part contains clearly defined enumerated list of tasks students have to carry out during the work.

This work is supported by the Thuringien Ministry of Science, Research and Art (Germany) and by EU V Framework project REASON. We also thank Jaan Raik for the Turbo Tester software development.

- *Example* – comprehensive and detailed description of what should be done at each step of the work. The tools which are used during the work described extensively in this part. We give also examples of representation final results.

3. THE WORKS

In this section we give basic information about the works. For more details, please, visit our dedicated web-pages [7]. At the moment the whole laboratory training consists of four available on-line laboratory works and two works on the stage of development. The following works are available on-line:

- Test Generation
- Design Error Diagnosis
- Built-in Self-Test
- Design for Testability

Test generation work introduces the basics of testing to students. The work provides with skills of composing simple tests manually as well as gives an experience in using of several basic automatic test pattern generators (ATPG). Hence, students can compare different test generation approaches and realize their suitability for different classes of devices or circuits.

The goal of the *design error diagnosis* work is to introduce the basics of diagnosis using the design error diagnosis problem as a reference. Let there is a specification and an implementation of a certain design, which are not functionally equal. The task of design error diagnosis is to obtain a diagnostic test in order to find the exact location of the error. During the work students learn how to compose diagnostic tests and see the important difference between diagnostic and fault-detection tests.

Built-in self-test (BIST) work illustrates issues of embedded testing devices. Students explore and compare various built-in self-test techniques such as BILBO (Built-In Logic Block Observer), CSTP (Circular Self-Test Path), and Hybrid BIST. Playing with certain parameters of the BIST devices, they learn finding best BILBO and CSTP architectures as well as best combinations of stored and generated patterns in Hybrid BIST.

Design for testability work shows students how important is to design a well-testable system. Some solutions for improvement of the testability of not 100% testable circuits are discussed. A set of modified devices with different degrees of testability is provided to show how simple means can significantly lighten the task of the tester. Students are asked to try different given configurations and see what fits better for which circuit.

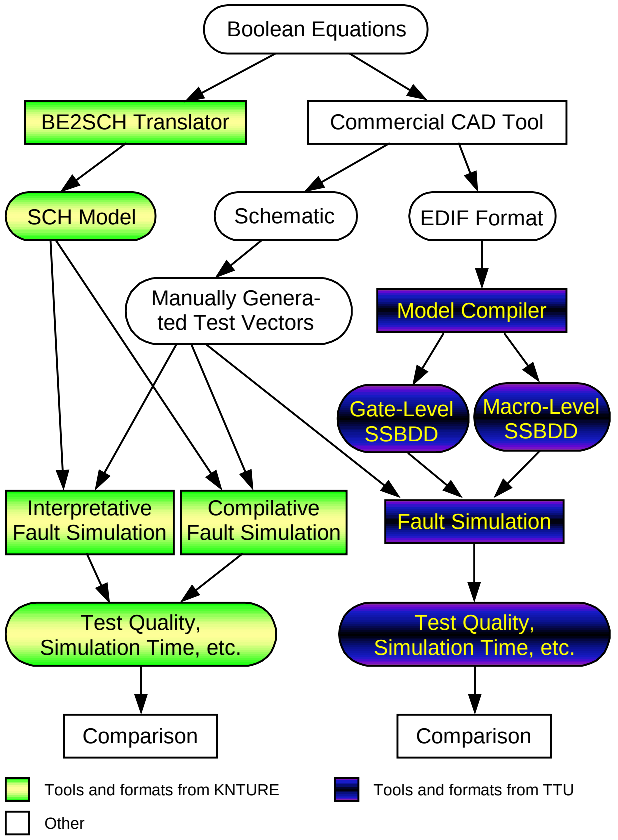
The described works are further discussed in [6]. They are based on the diagnostic software package Turbo Tester [3,4], which has been developed in Tallinn Technical University (TTU). The idea of another two works is a result of cooperation with Kharkov National Technical University of Radio Electronics (KNTURE). These two works take an advantage of joint software packages of both universities giving broader variety of testing tools and methods to learn. The mentioned works are:

- Fault Simulation
- RT-Level Test

The description of these works is accompanied by figures Fig.1 and Fig. 2. These figures show which software package is used for which part of the workflow. The bright boxes correspond to the tools and formats from KNTURE, while the dark ones represent the teaching system parts from TTU.

The idea of *Fault simulation* practical work (Fig. 1) is to illustrate different methods and approaches taken by engineers to speed-up the fault simulation process. In this work, students compare four methods of fault simulation for combinational circuits: interpretative vs. compilative and gate-level vs. macro-level. In this work we illustrate how different data models affects simulation speed.

Instead of commercial software and manually generated tests (upper white part in Fig. 1) for simpler version of this work a



Boolean Equations
BE2SCH Translator
Commercial CAD Tool
SCH Model
Schematic
EDIF Format
Manually Generated Test Vectors
Model Compiler
Gate-Level SSBDD
Macro-Level SSBDD
Interpretative Fault Simulation
Compilative Fault Simulation
Fault Simulation
Test Quality, Simulation Time, etc.
Test Quality, Simulation Time, etc.
Comparison
Comparison
Tools and formats from KNTURE
Tools and formats from TTU
Other

Fig. 1. The flow of the fault simulation work

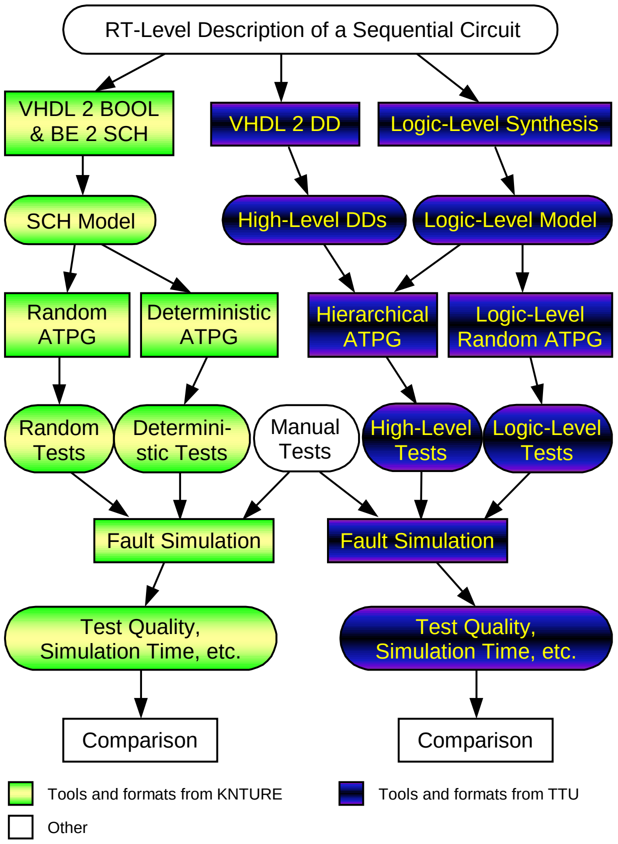


Fig. 2. The flow of the RT-level test work

set of pre-designed and pre-generated models can be prepared for downloading.

RT-level test (Fig. 2) is a work intended to introduce the basics of hierarchical testing as well as testing of sequential designs. The initial representation of the unit under test is the VHDL description. Using different synthesis tools, students generate different representation models, which allow to use different ATPGs: sequential random and deterministic algorithm based ATPGs as well as logic-level and a hierarchical one (which uses both high-level and gate-level models). The quality of those ATPGs is to be compared by students.

During this work students also learn manual high-level (functional) test generation. The quality of manual tests can be also compared to available ATPG approaches.

At the current moment, the laboratory training on design and test is covering all the main topics in testing and diagnosis of digital design. It is mostly based on the methods working at the logic and RT levels. The future development of the laboratory training can be directed to cover the specific problems of high-level verification and test of a SoC design.

4. PC-BASED TOOLS FOR TEACHING TEST

Traditional VLSI test generation and fault simulation software for workstations are both costly and unable to handle large numbers of students simultaneously in educational courses. During the recent years, many different low-cost tools running on PCs have been developed to fill this gap. They include usually the major basic tools needed for IC design: schematic capture, layout editors, simulators and place and route tools. In spite of this, low-cost systems for solving a large class of tasks from the dependability area: test synthesis and analysis, fault diagnosis, testability analysis, built-in self-test, especially for teaching purposes, are missing. For this reason, a diagnostic software Turbo Tester [3,4] has been developed in TTU and a similar system [5] - in KNTURE.

4.1. Turbo Tester

The Turbo Tester consists of various tools (see Fig. 3) for test pattern generation, simulation, test set optimization, BIST emulation, design error diagnosis, and experimental statistics representation. ATPGs and fault simulators are available for both, sequential and combinational circuits. Combinational circuits can be tested by deterministic, random and genetic algorithm based methods, while for sequential designs a random ATPG is available. In addition to fault simulators, the simulation tools include multi-valued simulation for hazard analysis in combinational circuits. For BIST emulation, BILBO and CSTP approaches can be chosen. For more details refer to the TT manual available at [4].

Another useful feature of the Turbo Tester is the ability to illustrate statistics in a graphical way. The following example (Fig. 4) uses test sequences generated by three different ATPGs and the BILBO emulator to visualize the speed of growth of progressive (cumulative) coverage of the mentioned sequences for c432 ISCAS'85 benchmark.

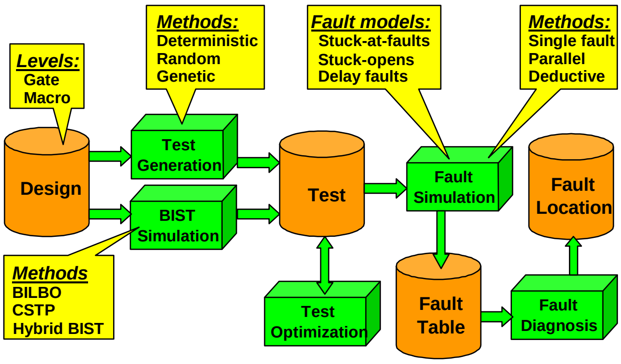


Fig. 3. PC-based diagnostic tool set Turbo Tester

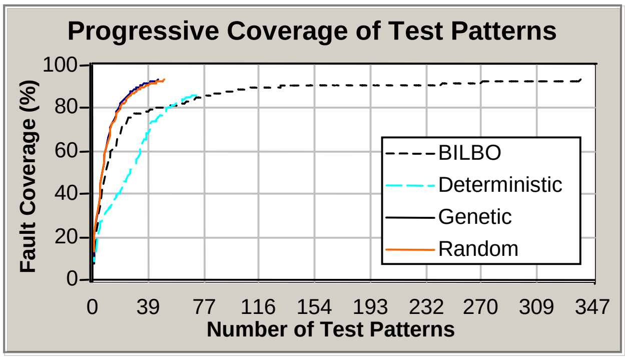


Fig.4 Progressive coverage of test patterns

The diagnostic software (Fig. 5) developed in KNTURE consists of the following three major parts.

4.2. Fault simulator

Fault simulator is intended for single stuck-at fault simulation of digital circuit, where the digital circuit is described at functional level in form of cubic coverages. The problems solved by the program: 1) single stuck-at fault simulation (SSF) simulation on cubic coverage of functional primitive elements, 2) simulation of complements to states of circuit lines on cubic coverage of functional elements, 3) algorithmic and pseudo-random test generation 4) test set compaction, 5) minimization of number of algorithmic generators that cover all SSFs in a circuit. The initial description of the circuit under test is VHDL.

The test generation program (TESTBUILDER and ASFTEST) is based on deterministic, algorithmic pseudo-random and genetic algorithm methods.

4.3. TESTBUILDER

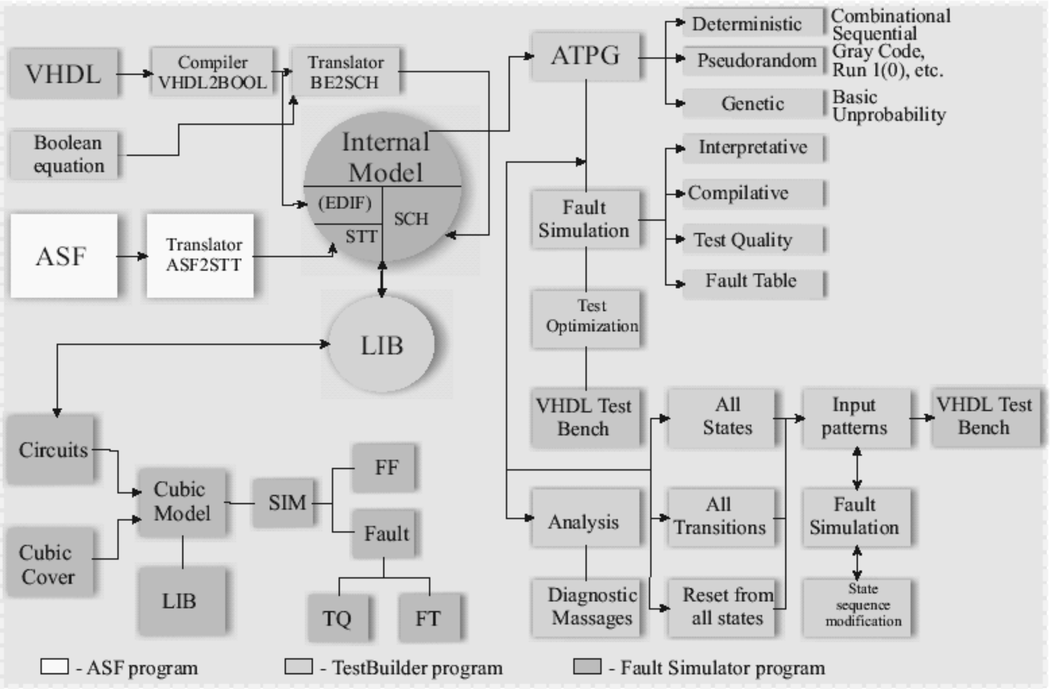
This program is intended for ATPG with respect to SSFs of digital designs described in a form of Boolean equations. The program functions are: 1) pseudo-random test generation by built-in binary and decimal code generators, 2) deterministic binary test-vector generation (by sensitizeing single signal paths in circuit), 3) single stuck-at fault simulation, 4) test formatting in the VHDL standard – Testbench.

4.4. ASFTEST

This program is an ATPG for finite state machines (FSM), where the initial information is represented in ASF format. This program is developed as a part of CAD Active-HDL. The program generates minimized test patterns in form of VHDL (or Verilog) file. These test patterns are used for design validation and verification by a CAD tool. Three strategies of test generation are possible: 1) the minimized traversal of all FSM graph nodes by solving problem of Euler circuit building, 2) the minimized traversal of all graph arcs by Hamilton circuit definition, 3) the FSM setting into initial state after the next node reaching. The indicated strategies are chosen depending on purposes of simulation and need for better test coverage.

5. SOME EXAMPLES OF WORK WITH PC-BASED TOOLS

Having the wide spectrum of different tools it is good to start practicing with test generation. One of the tasks formulated in the practical works is application of several test generation



TQ - Test Quality; FT - Fault Table; STT - State Transition Table; SCH - internal format

Fig. 5. PC-based diagnostic software from Kharkov National University of Radio Electronics

approaches to an 8-bit full ripple-carry adder and then the comparison of acquired results. In Table 1 we give the results of test pattern generation by three different ATPGs and by hand.

The table shows the most important parameters of a test pattern generation method such as the number of generated test patterns and test generation time for two modes of test generation. The first mode is called *default mode* when all the ATPG parameters are set with default values. Usually it implies the minimization of test pattern generation time. And we can see that among all the ATPGs the fastest is the deterministic one but in terms of the number of patterns the best is the genetic ATPG. However, neither of them could achieve as best results as we got manually. So, now we are trying to change some of the parameters of the ATPGs (not the deterministic because the one we use here cannot be tuned to obtain shorter test) in hope to get better results. These results are given in the “tuned” columns of the table. You can see that the test sequences are already very short but still not as best as that, which we composed by hand. It is a good example, which illustrates that sometimes if we know the functionality of the unit under test, we can compose more efficient tests than those generated by an ATPG.

6. CONCLUSIONS

A conception is presented for improving the skills of students to be educated for hardware and SOC design in test related topics. Advanced laboratory courses have been introduced with the goal to teach and train students to integrate design and test, and to give them knowledge on how to create testable designs or designs with self-testing capabilities and how to obtain test patterns of better quality.

TABLE 1 Comparison of several TPG methods

Test Generation Method	Default		Tuned	
	# of Patterns	Time, s	# of Patterns	Time, s
Deterministic ATPG	20	0.01	*N/A*	*N/A*
Genetic ATPG	13	4.33	9	70.18
Random ATPG	24	0.26	9	6.74
Manually	8	*N/A*	*N/A*	*N/A*

The tasks chosen for the training represent simultaneously real research problems. This allows to foster in students critical thinking, problem solving skills and creativity in a real research environment and atmosphere.

The course is successfully implemented and run gaining the increasing popularity among students. The tools and course have been tested also internationally and have received good credits from students of Michigan State University in USA, of Jonköping University in Sweden, of Helsinki University of Technology in Finland, from design engineers in Sweden trained by the company DIGSIM DATA AB.

REFERENCES

[1] M.L. Bushnell, “Increasing Test Coverage in a VLSI Design Course,” in *Proc. ITC*, Atlantic City, USA, 1999, p. 1133

[2] V.D. Agrawal, “Increasing Test Coverage in a VLSI Design Course,” in *Proc. ITC*, Atlantic City, USA, 1999, p. 1131

[3] G.Jervan, A.Markus, P.Paomets, J.Raik, R.Ubar, “A CAD System for Teaching Digital Test,” in *Proc. European Workshop on Microelectronics Education*, Kluwer Academic Publishers, Noordwijkerhout, the Netherlands, 1998, pp. 287-290.

[4] Turbo Tester home page URL: http://www.pld.ttu.ee/tt

[5] V. Hahanov, A. Babich, I. Sysenko, “System for Digital Device Test Generation in Active-HDL”, in *Proc. MIXDES Conf.*, Zakopane, Poland, June 21-23, 2001, pp. 235-238.

[6] A. Jutman, R. Ubar, “Laboratory training for teaching design and test of digital circuits,” in *Proc. MIXDES Conf.*, Zakopane, Poland, June 21-23, 2001, pp. 521-524.

[7] Lab training URL: http://www.pld.ttu.ee/diagnostika/labs/

[8] F. Brglez, H. Fujiwara, “A neutral netlist of 10 combinatorial benchmark circuits and a target translator in FORTRAN,” *ISCAS, Special Session on ATPG and Fault Simulation*, 1985.