Internet-Based Software for Teaching Test of Digital Circuits

R. Ubar, E. Orasson, H.-D. Wuttke

Abstract - A new teaching concept for teaching testing issues in digital design, which supports the possibility of distance learning as well as a web-based computer-aided teaching is presented. It offers a set of tools ("interactive modules") to inspect the teaching topics and to carry out laboratory research. The interactive modules are focused on easy action and reaction, learning by doing, a game-like use, and fostering students in critical thinking, problem solving skills and creativity.

I. INTRODUCTION

The rapid developments in the areas of deep-submicron electron technology and design automation tools are enabling engineers to design larger and more complex integrated circuits. This progress is driving engineers towards design methodologies called System-on-Chip (SoC).

The more complex are getting electronics systems the more important are getting the problems of test and design for testability, as the expences of verification and testing are becoming the major components of the design and manufacturing costs of new electronic products. The design and test cannot be seen any more as separate engineering issues. The emphasis on the quality of products, coupled with the growing complexity of electronic systems to be designed, require testing issues to be considered early in the design process.

At present, most VLSI and system designers know little about testing. The 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. This reflects the today's university education: everyone learns about design, but only truly dedicated students learn test [1-3].

Entering into the SOC era means that test must 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, and trained in test technology to enable them to produce high quality and defect-free products.

Teaching the basics of Digital Test and Testable Design means to teach a lot of complex connections that have to be explained at first one by one but then in their dynamic interactions. Traditional teaching methods using desk, overhead or "PowerPoint™" presentations can explain those connections only partially.

The students mostly get some accompanying materials such as scripts, books etc. After the lecture the dynamic part of the lecture, the connections created by the teacher between different subjects get lost and the students only have a static part of the whole scenario in their notices. After listening to a lecture they can consult only their notes and try to solve some problems by using the new learned method as good as they remember.

An enhancement of this situation can be reached, if the whole material, containing all scenes of the dynamic process can be accessed via the Internet.

In the following a conception and tools are presented to increase the teaching quality in the field of electronics design and test. The main target is to improve the skills of students to be educated for hardware and SOC design in test related topics. The work is a result of a cooperation carried out in 1999-2001 between Technical University of Ilmenau (Germany) and Technical University of Tallinn (Estonia) under the project "DILDIS" supported by the Ministery of Education in Thüringen.

II. LEARNING DESIGN AND TEST WITH "LIVING PICTURES"

The core of the teaching concept presented here are JAVA-applets (interactive modules) running on any browser connected to the Internet. We call this type of applets - "Living Pictures" [4,5]. The applets simulate tricky, quite complicated situations of the learning subject in a graphical form on the computer. The graphic is self-explanatory and provides interaction possibilities. By using these possibilities the students can generate examples that are interesting enough to encourage their own experiments but not too complicated for learning.

The program for representing "Living Pictures" for teaching Digital Test is written in Java 1.3. It can be run over network, using standard browsers like Netscape and Internet Explorer with Java 1.3 runtime plug-in, or with Java 1.3 applet viewer. The program can be used for teaching the basics of testing digital systems, test generation, fault simulation and fault diagnosis.

The work window of the applet consists of three main parts (Fig. 1):

- Vector insertion panel
- Design schematics panel

R.Ubar and E.Orasson are with the Department of Computer Engineering, Technical University of Tallinn, Raja 15, 12618 Tallinn, Estonia, E-mail: raiub@pld.ttu.ee

H.-D.Wuttke is with Technical University of Ilmenau, D-98684 Ilmenau, Germany, Dieter.Wuttke@theoinf.tu-ilmenau.de

- Data panel for displaying information (data tables and waveforms).

Vector insertion panel has two sub-panels: sub-panel I for inserting single input test vectors, and sub-panel S for setting up the feedback configuration of a Linear Feedback Shift Register (LFSR) to be used for generating pseudorandom test vectors automatically. In the LFSR mode, the first sub-panel is used for initializing the test generator.

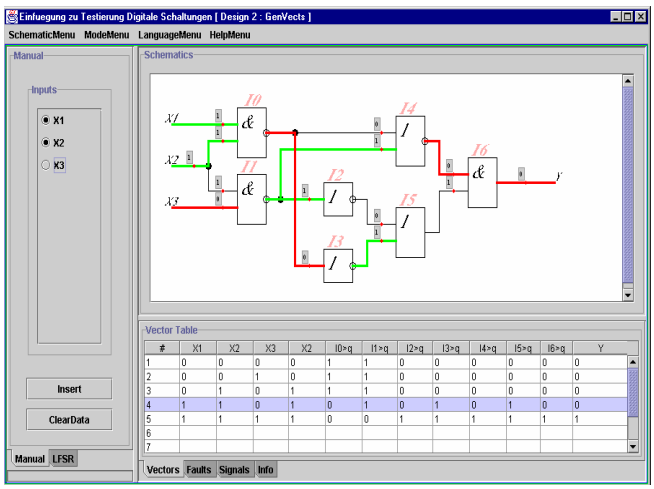

Fig.1. Java Applet for teaching Digital Test

The LFSR based Automated Test Pattern Generator (ATPG) is used for emulating different BIST ideas. The following well known architectures can be emulated: Built-In Logic Block Observer (BILBO) or Circular Self-Test Path (CSTP) [6,7].

Design schematics panel displays currently selected schematics. The small boxes at the lines display internal signal values on the connections. The boxes are intended for point-and-click during manual test vector generation and fault diagnosis. In the test generation mode, the needed signal values for fault activation or fault propagation can be inserted directly at the connections. In the fault diagnosis mode, by clicking the boxes, a guided probing procedure can be simulated. A click on the box shows the result of measuring the "real" signal on the corresponding connection of the simulated faulty circuit.

Data panel displays information about simulated test vectors and detected faults. In the fault simulation mode one can click on the row of a given test vector and have a visualization which faults are detected by this vector. In the signal (waveform) mode one can select all the signals of interest and leave out those which are not.

Menues. The design schematics menu contains a list of predefined circuits. By the language menu the user may choose one of the currently supported languages from the given list. The help menu provides with useful tips and explanations. The mode menu tells the applet what is to be done - test vector insertion, manual test vector generation, fault simulation or fault diagnosis (two modes are possible: sequential and combinational diagnosis).

One starts working with the applet by selecting in the design schematics menu a circuit from a set of predefined designs. Then different experiments with the chosen circuit can be carried out by selecting a proper working mode from the mode menu.

Vector insertion. In the vector insertion mode one can choose test vectors either automatically by using LFSR, or manually by inserting vectors on the subpanel I. In the manual mode, input patterns are inserted bit by bit, and simultaneously simulated. The boxes at the lines on the design schematics panel display the results of simulation – the values of signals on connections.

When using LFSR, one has to specify the initial state on the sub-panel I, to set up the feedback structure on the sub-panel S, and to specify the length of the test sequence. In the BILBO mode, a pseudo-random test sequence is generated by LFSR based on the settings on the sub-panels I and S. In the CSTP mode, the pseudo-random test sequence is generated by the composition of the LFSR and the circuit selected.

Test generation. In the test generation mode one chooses a target fault in the design and creates step by step the proper activated paths in the circuit. Activization of the fault and propagation of the error signals caused by the fault towards output are carried out by clicking the needed values into boxes on the internal lines of the circuit. From these values, finally, an input test vector will be deduced. The colors on lines help to understand the current status of the task: activated faults and activated paths are marked by red and green lines, the inconsistencies of the signal values are highlighted by blue color. The faults, detected by the test vector, can be displayed also on the data panel in form of a row in the fault table.

Fault simulation. In the fault simulation mode, a fault table is generated and shown on the data panel for all the test vectors created at the given moment. By selecting a vector on the data panel, all the faults detected by this vector will be highlighted by colors on the design schematics panel.

Fault diagnosis. In the fault diagnosis mode at first, the fault table should be generated by running the fault simulator for the set of previously generated test vectors. Entering into the diagnosis mode will insert a random fault into the circuit.

The following diagnosis strategies can be investigated: combinational and sequential diagnosis. For learning the combinational strategy, a single vector or a subset of vectors can be selected and applied to the erroneous circuit (imitating test experiments). The applet shows the results of testing, and displays the subset of suspected faults as well. To improve the diagnostic resolution, additional test vector(s) may be generated and used in the repeated test experiment.

Sequential diagnosis is based on the guided probing strategy. A test pattern is applied and the expected behavior of the circuit is displayed. By clicking on the connection

boxes the "real" values of signals of the faulty circuit can be measured.

The main didactic point in learning diagnostic strategies is in trying to localize the faults by as few test vectors (in the combinational approach) or by as few measurements (in the case of sequential approach) as possible. In this task a competition between students can be carried out which akes the "play" with the applet even more exciting.

The described program can be used for teaching the basics of testing digital systems. The teacher can use the applet during the lecture explaining the different problems of test generation, fault simulation and diagnosis. The applet also can be used during the exam for giving some tasks to students. Students can use the same applet for training purposes. They can insert different possible faults, and watch how the faults change the circuit's behavior at different input patterns, how the test patterns can be generated to detect a given fault, or how the faults can be localized by test patterns.

After acquiring some experience in these topics by "playing" with applets for relatively small and trivial pre-designed circuits, the students can start laboratory experiments on larger circuits designed by themselves or given by the teacher. For that purpose a set of low-cost PC-based tools for training digital test has been developed at TU Tallinn [8,9].

III. PC-Based Tools for Training Test

The main functionalities of the software package proposed for teaching test and design for testability are: diagnostic model generation, test pattern generation, fault simulation, testability measuring, built-in self-test quality evaluation and fault diagnosis (Fig. 2).

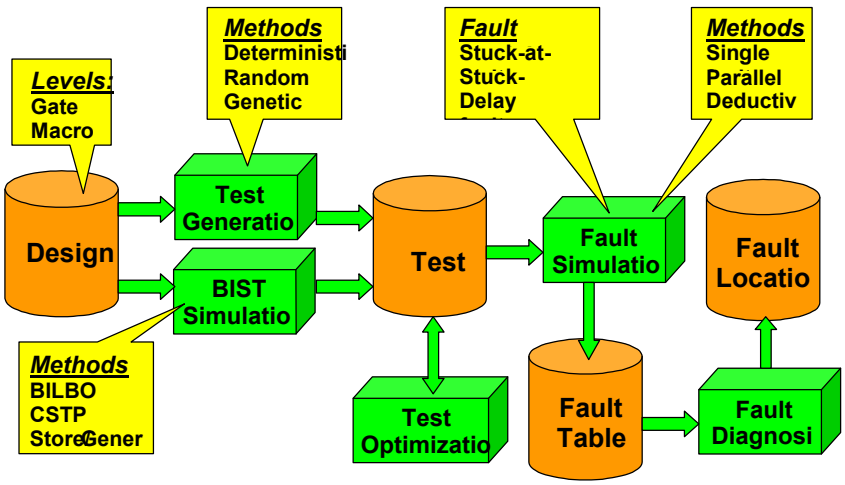

Fig.2. PC-based tool environment for training design for testability

The main didactic advantage of the system lies in the fact that different methods and algorithms for the basic test solutions are implemented and can be investigated and compared.

Model Synthesis. The component library consists of binary decision diagram (BDD) representations for the library components of the circuits to be processed. The library is open and can be updated for new components. From the netlist of the design, produced by schematic editor, the model generator creates a BDD-representation of the design. The design can be represented either at the gate-level or at the macro-level. On the macro-level, to each macro, a BDD will be created where one-to-one correspondence between signal paths in the macro and nodes in the BDD will be established. Two possibilities for representing the circuit give the possibility to investigate how the productivity of tools is depending on the model complexity.

All the tools implemented use the BDD-representation as the only information about the circuit to be processed.

Test Generation. In the system, random, deterministic and genetic test pattern generators (TPG) are implemented. Mixed TPG strategies based on different methods can also be investigated. Tests can be generated for both, combinational and sequential circuits. Stuck-at faults and transition faults can be considered. The number of faults to be processed at the macro level will be less than the number of faults at the gate level (each macro-level fault represents, in general, a subset of gate-level faults). This causes that the productivity of test generation at the macro level will increase compared to that of the gate-level.

Test Pattern Analysis. In the system, single-fault simulation, parallel fault simulation and critical path tracing fault analysis methods are implemented. These competing approaches can be investigated and compared for circuits of different complexities and structures, which gives to students better understanding of the essence of fault simulation problems.

As the result of using these tools, fault tables and fault coverages for the given test sequences will be calculated.

Testability Analysis. The true cost of a digital product is expressed as: Cost (Design + Test) < Cost (Design) + Cost (Test). From this, it follows that the total product cost can be minimized by regarding the design and test of a product as one integral activity rather than two disjoint, unrelated activities, which is called design for testability (DFT). Among the most promising DFT methods are those aimed at enhancing the testability through adding redundant hardware elements or test-points (additional outputs for observing, inputs for controlling, additional flip-flops in scan-path etc.) to the circuit. The testability analysis tools of the system can be used for detecting principially not testable faults, for selecting statistically hard-to-test faults, and for estimating the controllability, observability and testability characteristics for the nodes of the design. The tools are used for finding out where to alter the design to improve the testability.

Built-In Self Test (BIST) Quality Evaluation. The BIST approach is represented by applications for Built-In Logic Block Observer (BILBO) and Circular Self-Test Path (CSTP) emulation. Different BIST architectures can be simulated and the self-test quality of these architectures can be evaluated. It is possible to use also the general "store-and-generate" approach [6], where the whole test will be generated on the basis of a given set of test patterns (i.e. the stored part of the test). All these patterns will serve as initial input test patterns for on-line test generation by BILBO or CSTP (i.e. the generated part of the test).

The described set of tools can be installed under MS Windows/NT and Solaris operating systems. The tools can read the schematic entries of various contemporary VLSI CAD tools, e.g. Cadence, Synopsys, Mentor Graphics, Viewlogic, Compass, OrCAD, etc., which makes the system open to different design environments. Many of the commercially available and in-house test design systems usually have problems with the design interface.

Similarly to the Java-based "Living Pictures", the described test synthesis and analysis tools are accessible over the Internet [9]

IV. Conclusions

By the use of web-based media we achieve: presentation of course material independent of place and time, individual learning according to the students' own needs, new forms of communication between teachers and students, up-to-date course material.

The conception presented allows to improve the skills of students to be educated for digital hardware and SOC design in test related topics. It is a combination of learning the topic by using Internet based simple "Living Pictures" on one hand, and hands-on training by using a set of commercial design tools and low-cost university tools dedicated for simulating and estimating different test and testability solutions on the other hand. The tasks chosen for hands-on training simultaneously represent real research problems, which allow to foster students in critical thinking, problem solving skills and creativity in a real research environment and atmosphere.

On the basis of the described software at the Tallinn Technical University 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. The courses have gained a great popularity among students. The tools and courses 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.

Acknowledgements. This work is supported by the Thuringien Ministry of Science, Research and Art (Germany), by EU V Framework project REASON, and by the Estonian Science Foundation (grants No 3658 and 4300). We also thank Jaan Raik for the Turbo Tester software development, and Artur Jutman for setting up the laboratory course, and many other students involved in this development.

References

[1] Bushnell, M. L., "Increasing Test Coverage in a VLSI Design Course", *International Test Conference*, Atlantic City, NJ, USA, 1999, p. 1133.

[2] Harrington, J., "VLSI Design 101 – the Test Module", *International Test Conference*, Atlantic City, NJ, USA, 1999, p. 1134.

[3] Agrawal, V. D. "Increasing Test Coverage in a VLSI Design Course", *International Test Conference*, Atlantic City, NJ, USA, 1999, p. 1131.

[4] Wuttke, H.-D., Henke, K., Peukert, R., "Internet Based Education - An Experimental Environment for Various Educational Purposes". *Proc. of the IASTED Int. Conf. on Computers and Advanced Technology in Education*, Philadelphia, PA USA. IASTED/ Acta Press No. 292, 1999, pp. 50-54.

[5] Ubar, R., Wuttke, H.-D., "Action Based Learning System For Teaching Digital Electronics And Test" *,3rd European Workshop on Microelectronics Education" (EWME 2000), Aix-en-Provence, Kluwer Academic Publishers,* May 18-19 2000, pp. 107-110.

[6] Abramovici, M. et al., "Digital Systems Testing and Testable Design" IEEE Press, 1999, p. 652.

[7] Bushnell, M. L. Agrawal, V. D., "Essentials of Electronic Testing for Digital, Memory and Mixed- Signal VLSI Circuits". *Kluwer Academic Publishers*, 2000, pp 690.

[8] Ubar, R., "Teaching Dependability Issues in System Engineering at the Technical University of Tallinn", *Global J. of Eng. Education*, Vol.2, No 2, 1998, pp. 215-218.

[9] http://www.pld.ttu.ee/tt/