OVERVIEW OF E-LEARNING ENVIRONMENT FOR WEB-BASED STUDY OF TESTING AND DIAGNOSTICS OF DIGITAL SYSTEMS

A. JUTMAN[1], R. UBAR[1], H.D. WUTTKE[2]
[1]*Tallinn Technical University, ATI, Raja 15, 12618 Tallinn, Estonia*
[2]*Technical University of Ilmenau, Helmholtzplatz 1, 98693 Ilmenau, Germany*

1. Introduction

In this paper, we present an overview of latest developments taking place at Tallinn Technical University (TTU) in the area of e-learning supported by European project REASON (REsearch And Training Action for System On Chip DesigN) [8]. The environment consists of several interrelated modules shown in Figure 1. The PC-based tools installed locally and Java applets invoked remotely via Internet supplement each other and form the engine of the whole concept. The PC-based tool set is called Turbo Tester (TT) [1,7] and consists of the following main tools: test generators based on various algorithms, logic and fault simulators, a test optimizer, a module for hazard analysis [6], a simulator and test generator for defects [2], built-in self-test simulators [4], design verification and design error diagnosis tools [5]. This range of compatible diagnostic tools forms, via their interaction and complementary operation, a homogeneous research environment, which provides good possibilities for laboratory training and experimental research.

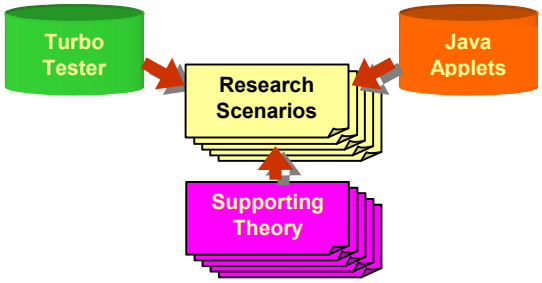

Fig. 1. Overview of the e-learning environment

The general idea behind the Java applets is a bit different. They mainly aimed at supporting the concept of game-like style of learning via easy action and reaction, learning by doing, and concentration on most important topics in the simplest possible way. There are three applets available by now [9]. They are described in this paper. The fourth applet, which represents a simple schematic and decision diagram (DD) editor, is currently under development.

The central point of the presented concept is research scenarios – a set of problems and experiments that represent virtual laboratory works, where students learn diagnostic software and get acquainted with common concepts and problems of testing and diagnostics. There are scenarios for beginners and for advanced study. The user-friendly graphical environment created by the applets is the best option for beginners. More advanced study is possible with scenarios, which make use of TT tools. The full text of the scenarios is available in the Web [10].

The rest of the paper is concentrated on the description of the applets and related scenarios leaving the PC-based part of the e-learning environment behind the scope of the article.

2. Overview of Java Applets and Related Work Scenarios

There are three applets already available in the Web [9] and one more is currently under development. Figure 2 represents an overall structure of relations between the applets, Turbo Tester, and the research scenarios. As it is seen from the figure, the scenarios form two distinct groups: Java applet-based (Web-based) ones and TT-based (advanced) ones. The first group of scenarios fully exploit the functionality of the corresponding applet and therefore each such applet-scenario pair represents a self-contained system aimed at teaching target area of knowledge and engineering. We tried to design our applets in a uniform way; so that the user once got acquainted with the overall style does not have to spend his time learning the new style once again from the beginning. In the following we describe the functionality of the applets in more detail.

Java applet on basics of test & diagnostics. Main principles of logic-level test generation and fault diagnosis are the fundamentals of the modern VLSI CAD and diagnostic software. Good understanding of these basics gives students vital background and skills in their future profession. The applet provides with possibility of manual and pseudo-random test pattern generation (TPG), fault simulation, combinational and sequential fault diagnosis, and investigation of BIST techniques. All the designs used in the applet are combinational circuits of a rather small size represented and visualized on the logic level.
In the test generation mode one chooses a target fault and step by step activates needed paths in the circuit in order to detect the fault. At the same time, the lines (wires) change their color helping the student in selection of proper values. If the specific fault detection is not of a primary interest, the user can specify the input values only. In this way one can set up as many vectors as he wants. The pseudo-random TPG is also implemented. It provides two modes: BILBO and CSTP [3]. All the vectors can be then simulated in the fault simulation mode. The results of simulation are represented in the form of fault table. By selecting a vector in the table, all the faults detected by this vector will be highlighted on the design schematic. In the combinational fault diagnosis mode, a subset of vectors is

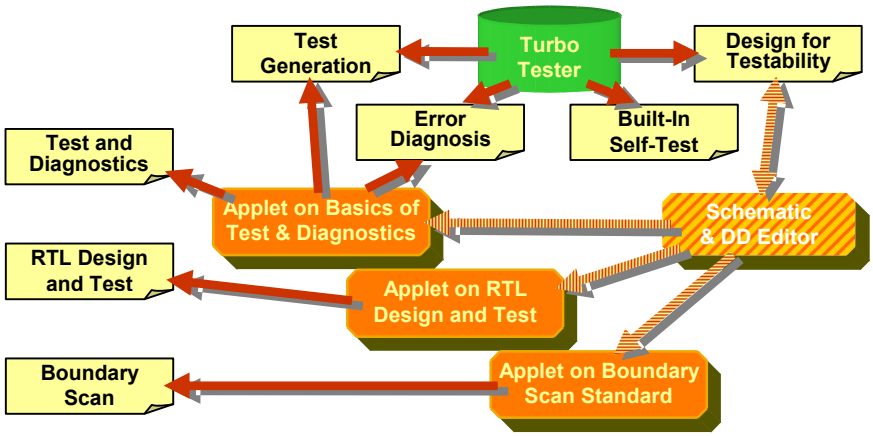

Fig. 2 Relationship between the applets, TT, and research scenarios

selected and applied to the erroneous circuit (imitating test experiments). The applet shows the results of fault diagnosis by highlighting the faulty area. If the results are not satisfactory, additional vectors could be generated.
Another supported method of fault diagnosis, the sequential diagnosis is based on the guided probing strategy. In this mode, students should sample the values by clicking signal lines and comparing observed values and correct ones. The goal is to find the precise fault location using as few samples as possible.

Java applet on RT-level design and test. In this applet, we combine and illustrate many different problems related to RT-level control intensive digital design, test, and design for test. Therefore, the applet gives a unique possibility to teach these topics in a consecutive iterative approach. The range of considered problems includes: design of data path and control part (microprogram) on RT level; investigation of tradeoffs between the system speed & HW cost; RT-level simulation and validation; gate-level deterministic test generation for RT-level functional blocks, functional testing, fault simulation, design for testability, logic and circular BIST, functional BIST, etc.
The applet provides the representation of the target system on RT level as divided into *datapath* and *control unit*. The structure of datapath is shown schematically, while the *control part* of the system is defined by the *Microprogram table*. The RT-level fault and fault-free simulation can be performed for a single set of input data as well as for all the sequence of input operands at once. During the simulation the active line of the microprogram is highlighted. The simulation data is also reflected in a graphical way. There is a manual test patterns generation mode for a selected microoperation separately. The gate-level representation of this microoperation is shown at that time. The test access to the selected unit is provided by a special *Test Microporogram*. There are various possibilities to experiment with BIST provided in the *BIST module*. The user can select locations of pseudo-random TPGs and signature analyzers within the data path as well as their configurations. The applet has a flexible design. The RT-level system model is described in a form of text-files. Hence, any custom design can be described and loaded just as simple as the original ones. The applet has a built-in extendable collection of examples implementing different algorithms. For connecting the applet to other applications as well as for allowing users to save the results of their work, the applet has a data import/export capability. It also has a built-in multilingual support.

Java applet on Boundary Scan (BS) standard IEEE 1149.1. There are two different supported modes of BS simulation. The first one, the *TAP Controller Mode*, provides a very detailed illustration of operation of BS registers and the TAP controller. This mode is intended for the beginners and for teachers. It helps to understand all the needed basics. Another mode, the *Command Mode*, can be used for faster simulation of BS commands like EXTEST, SAMPLE/PRELOAD, etc. with different predefined input data. This mode is useful for *fault diagnosis*. For that, the applet supports possibility of random or specific fault insertion. The operation of the faulty device can be then simulated and the fault can be diagnosed.
Our applet is provided with lots of built-in examples. Furthermore, it allows users to generate their own examples by creating fully custom chips or boards. In the chip editing mode, the applet reads the description of BS structures using BSDL

(Boundary Scan Description Language) format which is a part of the BS standard. Such BSDL descriptions are usually free of charge and widely available via Internet, which makes the work with the applet easier and more exciting, since the student can visualize behaviour of many different chips available in the market.

3. Conclusions

In this paper we have described Java applets used in the e-learning environment developed at Tallinn Technical University. This work is still in progress. All the components of the environment are available via the Web.
The core of the system is a set of research scenarios or virtual laboratory works based on two types of software: *a)* diagnostic package Turbo Tester and *b)* set of Java applets. While the TT should be installed on a local computer, the applets are invoked remotely via Internet. The Turbo Tester provides a homogeneous research environment, which allows for interesting experimental research to be conducted. The Java applets, in their turn, provide an attractive game-like milieu, which is important especially for beginners.
Due to the facts above, the conception, overview of which is presented here, is suitable for a broad audience of learners who are interested in studying different aspects of testing and diagnostics of integrated digital circuits.

4. Acknowledgements

This work was supported in part by the EU Framework V projects REASON and eVikings, by the Thuringian Ministry of Science, Research and Art (Germany), and by the Estonian Science Foundation Grants No. 5649 and No. 5910.

References

[1] M.Aarna, E.Ivask, A.Jutman, E.Orasson, J.Raik, R.Ubar, V.Vislogubov, H.-D.Wuttke. “Turbo Tester - Diagnostic Package for Research and Training,” *in Scientific-Technical Journal "Radioelectronics & Informatics".* KNURE. Vol. 3(24), 2003, pp.69-73.

[2] M. Blyzniuk, FT. Cibakova, E. Gramatova, W. Kuzmicz, M. Lobur, W. Pleskacz, J. Raik, R. Ubar. “Hierarchical Defect-Oriented Fault Simulation for Digital Circuits,” *IEEE European Test Workshop*, Cascais, Portugal, Mai 23-26, 2000, pp.151-156.

[3] M.L. Bushnell, V.D. Agrawal, *Essentials of Electronic Testing for Digital Memory and Mixed-Signal Circuits*, Kluwer Academic Publishers, Dordrecht: 2000, p. 690.

[4] G. Jervan, Z. Peng, R. Ubar. ”Test Cost Minimization for Hybrid BIST,” *IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems.* Tokio, October 25-28, 2000, pp.283-291.

[5] A. Jutman, R. Ubar, "Design Error Diagnosis in Digital Circuits with Stuck-at Fault Model," *Journal of Microelectronics Reliability*. Pergamon Press, Vol. 40, No 2, 2000, pp.307-320.

[6] R. Ubar. “Dynamic Analysis of Digital Circuits with Multi-Valued Simulation,” *Microelectronics Journal,* Elsevier Science Ltd., Vol. 29, No. 11, Nov. 1998, pp.821-826.

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

[8] REASON project home page: http://reason.imio.pw.edu.pl

[9] Java applets home page: http://www.pld.ttu.ee/applets

[10] Laboratory training URL: http://www.pld.ttu.ee/testing/labs