Asynchronous E-Learning Resources for Hardware Design Issues

Artur Jutman, Alexander Sudnitson, Raimund Ubar, Heinz-Dietrich Wuttke

Abstract: *The paper presents a conception of how to improve the skills of the students studying digital design and test related topics. A learning method based on using the so-called "living pictures" is applied. The goal of this method is to put interactive teaching modules on the Internet so that they can be used in asynchronous-mode learning of digital design and test. The system is implemented in the form of Java applets and can be freely accessed through the Internet. The principal message of the conception is to inspire students to learn, and to prepare them for developing problem-solving strategies.*

Key words: *Asynchronous-Mode Learning, Digital Design and Test, Java Applets.*

INTRODUCTION

The recent years have seen a rapid emergence and broad acceptance of distance learning technologies. These technologies can be divided into two categories: synchronous or asynchronous. Both terms describe a type of communication between the instructor and the learners. Although synchronous technologies (video teleconferencing, online chat and telephone conference calls) are very useful, the constraint of real time communication is very limiting for most distance learners. A more attractive alternative is asynchronous distance learning. Asynchronous distance learning is the form of distance learning where the communication between the instructor and the learners is not required to occur in real time. Web-based instruction is the most attractive form of asynchronous distance learning because it can incorporate synchronous and asynchronous technologies.

The core of the teaching system presented in this paper are several Java-applets running on any browser connected to the Internet. The use of Java applets can encourage asynchronous distance learning and thus overcome the limitations inherent in traditional instructional techniques. Java applets can help create an interactive environment of "leaning by doing". Beyond their ability to better convey certain concepts, the applets can increase motivation and instill greater interest among students.

LEARNING PROCESS

Our methodical approach is directed to the goal to introduce students step by step into hardware design topics. In every level of knowledge they get the possibility to repeat experiments demonstrated by the lecturer as well as to create own examples and to make own experiments. That's why we create already the applets for teaching basic knowledge in the form of "Living Pictures" [1, 2]. Typically, "Living Pictures" can be used like tools, supporting a design or test step. In that way students become familiar with self creation of examples and exploration strategies.

There are several phases of the learning process supported by the educational system we offer:

- the reading (or listening) phase;
- the replication phase (students can use the interactive worksheets from any computer connected to the Internet and they are able to gain their own experience with the modules);
- the examination phase (the interactive worksheets are a good summary of problems the solution of which are necessary to the test);
- the practice phase (students have to solve digital systems problems; to develop required logic design skills they can use the interactive worksheets as a set of tools supporting several phases of the process).

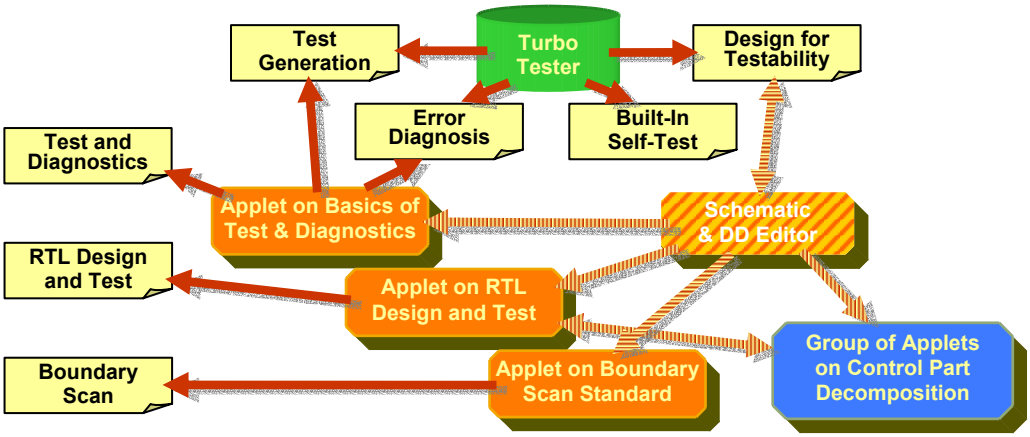

Fig. 1 Relationship between the applets on hardware design and test.

The learning process initially presents the knowledge of the domain and progressively enhances the learner's competence in the application of that knowledge in a working environment. For each phase, there exists a special application service allowing different views on actions. To implement the software system's architecture we should follow four main requirements:

1) possibility to run under various operating systems;
2) implementation of new modules without changing the rest of the system;
3) realizing a client-server architecture;
4) using the same source to generate worksheets to prevent inconsistency after modifications.

STRUCTURE AND CONCEPTS OF REALIZATION

The PC-based tool set is called Turbo Tester (TT) [3, 4] 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, a simulator and test generator for defects, built-in self-test simulators, design verification and design error diagnosis tools. 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.

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. The main features of "Living Pictures" conception incorporate: graphical representation of the learning subject, dynamic content, user-friendly interface, availability of various examples. Most of the applets are completed and available by now on-line [5], while some are still under development and testing phase.

The first large group of Java applets covers the microelectronics hardware design and test area. The second group of applets targets the problem of decomposition of final state machines (FSM). Figure 1 represents an overall structure of relations between these applets, the Turbo Tester, and the research scenarios. As it is seen from the figure, the scenarios form two distinct groups: Java applet-based (and therefore available on the Web) ones and TT-based (advanced) ones. Several advanced scenarios also make use of some functionality of the applets. Another 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. The java applets will be supplemented by a schematic editor that allows to edit circuits in different notation forms as for instance decision diagrams. It will close the methodical gap between the applets for teaching and the tools for research such as the Turbo Tester. Small designs can be explored first by the easy-to-use applets and after that exported to the tools.

The main part of the applet is the schematic view panel that provides the schematic representation of the target system. It reflects the internal structure of the data path, which is well re-configurable (different functions can be selected in each functional unit). The control part of the design should be specified in the micro-program table. This can be done in a relaxed way giving the possibility of designing many different devices using the same basic data path. Even the same device can be designed in many different ways using less or more of the hardware resources. This may result in a lower or higher speed of the system, which can also be measured.

CONTENT DESCRIPTION

The core of the teaching conception presented here is a combination of educational PC-based CAD software and a set of Java-applets that are available web-site [5].

Actually the java applets are supplemented by a schematic editor that allows to edit circuits in different notation forms as for instance decision diagrams. It closes the methodical gap between the applets for teaching and the tools for research such as Turbo Tester [4]. Small designs can be explored first by the easy-to-use applets and after that exported to the tools.

RT-level Design Applet

Entering the SoC era with its new concepts means teaching at higher levels of abstraction like register-transfer level (RT-level). The developed RT-level design and test applet allows to solve and illustrate many problems related to RT-level control intensive digital design and test.

The range of problems includes:

- design of data path and control path;
- investigation of trade-offs between speed and hardware cost;
- RT-level simulation;
- fault simulation;
- test generation;
- design for testability and Built-In Self-Test (BIST).

The system consists of the following parts:

- *Schematic View* panel provides the schematic representation of the design and the graphical simulation data;
- *Microprogram Table* panel is used to define the control path of the system (during the simulation this panel shows which part of the microprogram is currently executed);
- *Simulation and Test* tab-panels;
- *Simulation Results* tab-panel is the place where the results of simulation or test are stored;
- *Fault simulation* module provides fault simulation for the data path and its units;
- *BIST module* provides the basis to experiment with embedded self-test facilities.

For test generation either manually generated functional patterns or randomly chosen patterns (test data) can be used. For testing the blocks of the data-path, a special test microprogram can be implemented. The quality of tests can be estimated by gate-level fault simulation.

Fault simulation is carried out at the gate level. The process is controlled by the data in the microprogram table. The target of the fault simulation (a unit or the implemented microoperation in the unit) are selected by the student and then highlighted. The fault simulation data is reported as a fault table.

Two modes of BIST architectures are implemented: Built-In Logic Block Observer (BILBO) mode based on using random Test Pattern Generator (TPG) and Signature Analyzer (SA), or Circular Self-Test Path (CSTP) mode based on using combined TPG/SA scan-path register [6]. Both modes can be implemented in two ways: different settings for each combinational circuit to be tested, or the same setting for all circuits. The aim of the student's work is to find best settings.

The applet has a built-in extendable collection of examples implementing different algorithms. They help users to understand principles of the system operation. For connecting the system to other applications as well as for providing users with a possibility to save the results of their work for further use the applet has a data import/export capability.

Controller Decomposition

FSM is the oldest way of describing and modeling autonomous systems. Although design complexities have grown continuously, this classical description method is still used by several synthesis tools. FSMs have been widely used also to express algorithms, communication protocols, digital systems, sequential logic circuits, and sequential logic cells.

Decomposition has been a classic problem of discrete system theory for many years. A large hardware behavioral description is decomposed into several smaller ones. One goal is to make the synthesis problem more tractable by providing smaller sub-problems that can be solved efficiently. Another goal is to create descriptions that can be synthesized into a structure that meets the design constraints. In the past, synthesis focused on quality measures based on area and performance. The continuing decrease in feature size and increase in chip density in recent years have given rise to considering decomposition theory for low power as new dimension of the design process [7].

The theoretical background of our approach is the automata decomposition theory that uses partition pair algebra [8]. The importance of this theory lies in the fact that it provides a direct link between algebraic relationships and physical implementations of finite state machines.

For teaching decomposition research purposes, a set of applets for studying the basics of the decomposition theory of FSMs have been developed. The applet for FSM network construction allows to experiment with decomposition of the prototype machine. Different partitions can be chosen to decompose the given FSM to meet different design restrictions.

The developed applet can be considered as a research tool that we use to carry out experiments intended to further develop decomposition synthesis. Experiments can be carried out on a set of well-known FSM benchmarks.

Applet on Logic Level Test Generation & Diagnosis

The applet allows the test vector insertion by hands as well as automatic pseudo-random test generation by (LFSR). The first mode aims at illustration of different test generation approaches manually. In this mode, the needed signal values for fault activation or fault propagation can be inserted directly at the connections on the circuit

schematics making the process of test generation very illustrative. The LFSR mode is used for emulating different BIST architectures like (BILBO) and (CSTP) [9]. In the fault simulation mode, a fault table is generated for all the created test vectors. By selecting a single vector all the faults detected by this vector will be highlighted by colors on the circuit schematics. The applet allows insertion of stuck-at faults into the circuit. There are two different fault diagnosis modes possible. In the guided probing mode the fault diagnosis procedure can be simulated by clicking on the wires on the circuit schematics and "measuring" the "real" signal levels in the "defective" circuit. For learning the combinational diagnosis strategy, a single vector or a subset of vectors can be selected in the fault table (imitating test experiments). The applet simulates these vectors and shows the results of diagnosis displaying the subset of suspected faults. 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 makes the "play" with the applet even more exciting.

Java Applet on Boundary Scan Standard: Board Test

The testing of PCBs, as an important part of the manufacturing test requires new solutions. Some new very important standards have to be taught to future designers and test engineers. One of such standards is the Boundary Scan (BS), formally known as IEEE Std 1149.1 [4]. This standard is a set of design rules, which when applied at the chip level help reduce the cost of designing and producing ICs. The standard came about as a result of the efforts of a Joint Test Action Group (JTAG). The JTAG proposed basic test architecture to be incorporated at the IC level.

The applet allows several working modes: design/editing of the Boundary Scan structures inside the target chip using the BS description language (BSDL); design/description of the target board that consists of several chips; simulation of work of the TAP controller, scan register and other BS registers; insertion and diagnosis of interconnection faults. In the board description mode, each chip on the board can be defined and redefined.

The applet reads the description of BS structures using BSDL format and the description of the chip's internal logic in SSBDD format. Such BSDL descriptions are widely available for free via Internet. This makes the work with the applet easier and more exciting, since the student can visualize the work of many well known chips with BS available in the market The latter may also be interesting for test engineers. The simulation of the chip's work can be done in two modes. 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, helping to understand all the needed basics. Another mode, the command mode, can be used for faster simulation with different predefined input data and for the fault diagnosis. There is a possibility of random or specific fault insertion. The operation of the faulty device can be then simulated and the fault can be diagnosed.

CONCLUSIONS

This paper presents a new teaching concept with a consecutively method that combines training and research activities in a common environment. For teaching the use of tools dealing with simulation, test generation, fault simulation and fault diagnosis we developed a research training environment supporting different research scenarios. These scenarios support analytically as well as synthetically research. In the paper we present some examples of research directions and the conception of the research training environment.

We offer a set of tools which support the learning process in computer engineering area. As they are placed on the Web, every student or trainee throughout the world is able to gain access to these tools. On the one hand, teachers can demonstrate different examples and procedures of related topics using computer simulated living pictures during their lessons [4]. On the other hand, students can use the same simulations on their home computers.

We have selected the Java technology for our interactive teaching system since it is well supported by main operating systems like Windows, Linux, and Solaris. Java allows for graphical content to be easily created. It also has well developed means for creating the user interface. Furthermore, Java applets are running on any standard browser like Netscape and Internet Explorer connected to the Internet. The latter makes it easy for students, from universities all over the world, to use this system at any time and in any place.

ACKNOWLEDGEMENT

This work is partially supported by the Thuringian Ministry of Science, Research and Art (project DILDIS), by EU V Framework projects (IST-2001-37592 and IST-2000-30193), and by the Estonian Science Foundation (G5643 and G5649).

REFERENCES

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

[2] Ubar, R., H.-D. Wuttke. The DILDIS Project – Using Applets for More Demonstrative Lectures in Digital Systems Design and Test. 31st ASEE/IEEE Frontiers in Education Conference, . Reno, USA, October 10-13, 2001.

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

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

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

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

[7] Devadze, S., Fomina, E., Kruus, M., and A. Sudnitson. Web-Based System for Sequential Machines Decomposition. in Proc. IEEE EUROCON 2003 International Conference on Computer as a Tool, Ljubljana, Slovenia, 2003, V.1, pp. 57-61.

[8] Hartmanis, J., R. E. Stearns. Algebraic Structure Theory of Sequential Machines. Englewood Cliffs, N.J.: Prentice-Hall, 1966. [1] Ashar, P., Devadas S., and A.R. Newton. Sequential Logic Synthesis. Boston: Kluwer Academic Publishers, 1992.

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

ABOUT THE AUTHORS

Ph.D. Student Artur Jutman,
Assoc. Prof. Alexander Sudnitson, Ph.D.,
Prof. Raimund Ubar, Dr. Sc.,
Department of Computer Engineering, Tallinn Technical University, Raja 15, 12617 Tallinn, Estonia, Phone: +372 620 2251, E-mail: alsu@cc.ttu.ee
Prof. Heinz-Dietrich Wuttke, Ilmenau Technical University, Faculty of Informatics and Automation, post box 10 0565, D-98684 Ilmenau, E-mail wuttke@theoinf.tu-ilmenau.de