Java Applets Support for an Asynchronous-Mode Learning of Digital Design and Test

A. Jutman, A. Sudnitson, R. Ubar
Department of Computer Engineering
Tallinn Technical University
Raja 15, Tallinn 12618
ESTONIA
alsu@cc.ttu.ee

H.-D.Wuttke
Department of Information Technology
Ilmenau Technical University
Ilmenau
GERMANY
Dieter.Wuttke@theoinf.tu-ilmenau.de

***Abstract* – A system for teaching design and test of digital devices and systems at different levels of design flow is presented. The greater part of the system is intended mainly to illustrate register-transfer level problems in control intensive digital systems such as investigation of trade-offs between the system's speed and the cost of hardware, control part decomposition, simulation, fault simulation, test generation, built-in self-test, and some others. A conception of training system for teaching the IEEE 1149.1 Boundary Scan standard is presented. The system is implemented in the form of Java applets and can be freely accessed through the Internet. The latter makes it easy for the students to improve their learning records using the opportunities of asynchronous mode of education via the Internet.**

I. INTRODUCTION

The pace of progress in integrated circuits and system design has been dictated by the push from application trends and the pull from technology improvements. Entering the System-on-a-Chip (SoC) era is a real challenge for VLSI and system design courses. The more complex are the electronic systems, the more important are the problems of digital design and test at different levels of design flow [1], [2]. To cope with today's demands, the engineering curricula and learning technologies must be constantly updated.

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 [3].

In this paper 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.

The core of the teaching system presented 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 [4]-[6].

The paper is structured as follows. In Section II, we describe the main teaching concept of the system under consideration. In Section III, we describe an applet for teaching principles of Boundary Scan technique important for SoC design. Section IV is devoted to the register-transfer level (RT-level) design and simulation. Section V describes the applets developed for teaching some theoretical problems of the finite state machine (FSM) decomposition. Finally, some conclusions are presented.

II. GENERAL TEACHING CONCEPT

We apply the "learning by doing" paradigm in the Internet-based distance learning taking into account available computer and network resources.

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 good summary of problems the solution of which are necessary to the test);
- the practice phase (the students have to solve digital systems problems; to develop required logic design skills they can use the interactive worksheets like a set of tools supporting several phases of the process).

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 [4]:

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;

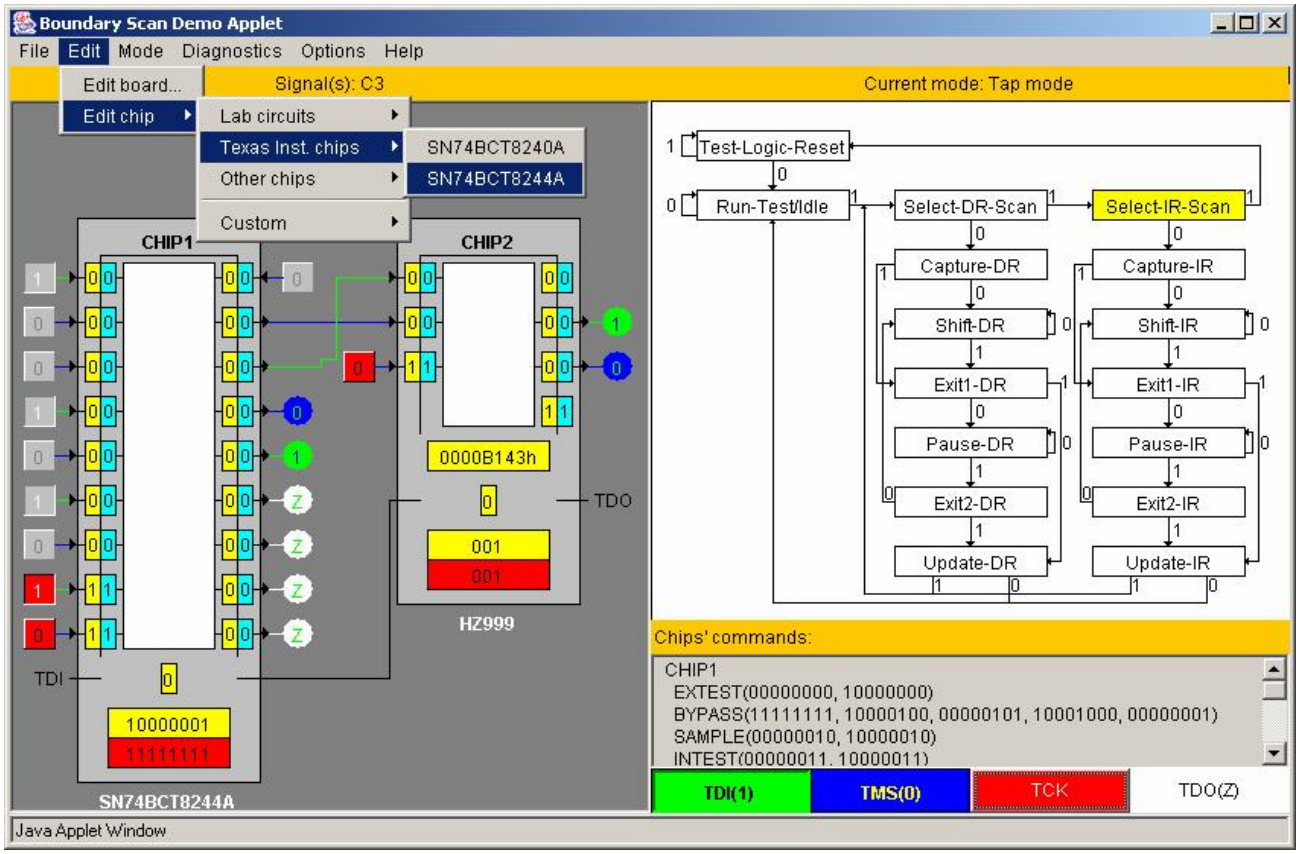

Fig. 1 Boundary Scan applet main window

4) using the same source to generate worksheets to prevent inconsistency after modifications.

These requirements determine the use the applet concept of the Java language. Java is a natural choice of the programming language by the client because of its flexibility of Graphic User Interface (GUI) design, convenient network programming, and platform independence. The latter is especially significant since it allows the same applet program to run on client computers of different platforms.

The foundation for the teaching concept presented here is a Java applet of a special type, which we call "Living Pictures" [4]. Those applets simulate tricky, quite complicated situations of the learning subject in a graphical form on the computer screen. The graphics 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.

In our teaching system [7] we succeeded to combine and illustrate many different problems related to control intensive digital design and test.

III. APPLET FOR LEARNING PRINCIPLES OF BOUNDARY SCAN TECHNIQUE

Printed Circuit Boards (PCBs) are the most valuable part of electronics hardware. Over the years, PCBs have become loaded with more components and hence have become increasingly complex and expensive. The testing of PCBs as an important part of the manufacturing test requires new solutions as well. There are some new, very important standards that have to be taught to future designers and test engineers. One of such standards is the IEEE Std 1149.1 "Test Access Port and Boundary-Scan Architecture" developed by Joint Test Action Group (JTAG): Interface system between electronic components, assemblies and systems, and external or built-in test equipment to provide those components, assemblies and systems with testability attributes [8]. A Boundary Scan (BS) device manipulation is quite a tricky exercise. Therefore, only a system, which allows instant simulation and illustration of all the student's steps can help learning and easy finding all possible mistakes and misunderstandings, which otherwise would likely be missed out. To learn this complex standard Texas Instruments company has developed a training system called ScanEducator [9].

All mentioned advantages of the Java environment are at the same time the advantages of our teaching system against the ScanEducator, which works under DOS only and must be installed locally instead of running over Internet. Another difference is that ScanEducator has only a couple of chips to work with, while our system is provided with a lot of built-in examples. We also decided to allow users to generate their own examples by creating a fully custom chip or board. Moreover, our applet has a specific fault insertion and diagnosis possibility.

The BS architecture implies the introduction of scan chains in such a way, that each pin of each chip receives an internal control point. The standard defines also the Test Access Port (TAP) and the TAP Controller [8]. All these structures do not seem that complicated if their operation is dynamically illustrated.

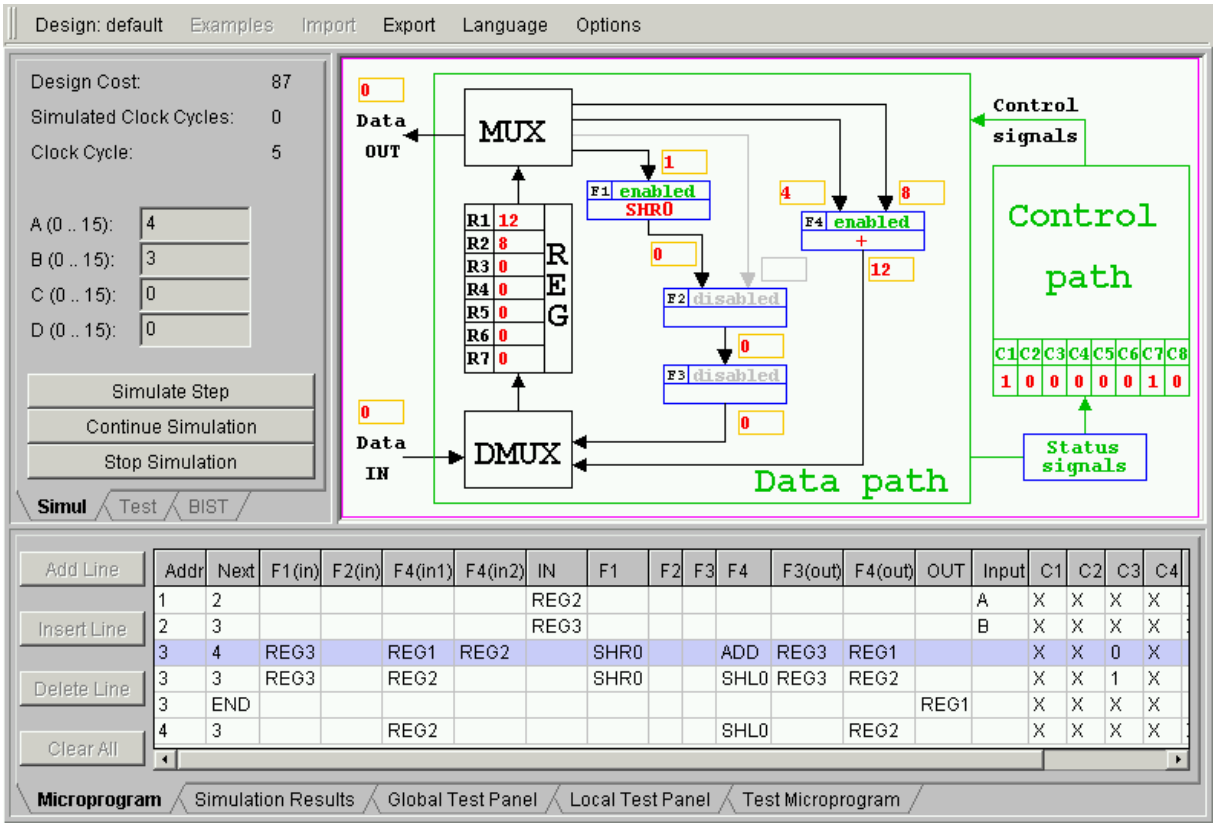

Fig. 2. RT-level design applet window

The applet (Fig. 1) allows several working modes:

- design/editing of BS structures inside the target chip using the BSDL language [8];
- design/description of the target board that consists of several chips;
- simulation of work of TAP Controller, scan register and other BS registers;
- insertion and diagnosis of interconnection faults.

In the Edit Board mode, each chip on the board can be defined and redefined. New chips can be also created and inserted. The applet reads the description of BS structures using Boundary Scan Description Language (BSDL) format, which is a part of the standard now. Such BSDL descriptions are widely available for free via the Internet. This makes the work with the applet easier and more exciting, since the student can visualize the operation of many well-known chips with BS available in the market. The latter may be interesting also for test engineers who need to check or debug their BS designs.

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 beginners and for teachers, helping to understand all the needed basics. The other 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.

IV. RT-LEVEL DESIGN AND TEST APPLET

RT-level is characterized by

- a digital system is viewed as divided into a data subsystem (data-path) and control subsystem (control-path);
- the state of the data-path is defined by the contents of a set of registers;
- the function of the system is performed as a sequence of register transfers (in one or more clock cycles);
- a register transfer is a transformation performed on a datum while the datum is transferred from one register to another;
- the sequence of register transfers is controlled by the control-path.

The RT-level design and test applet allows to solve and illustrate many problems related to RT-level control intensive digital design and test [2].

The range of problems includes:

1) design of a data-path and control part;
2) investigation of trade-offs between speed and hardware cost;
3) RT-level simulation;
4) fault simulation;
5) test generation;
6) design for testability and BIST (Built-In Self-Test).

The system (Fig. 2) consists of the following parts:

- *Schematic View* panel provides the schematic representation of the design and the graphical simulation data. The structure of the *data-path* is reflected there.

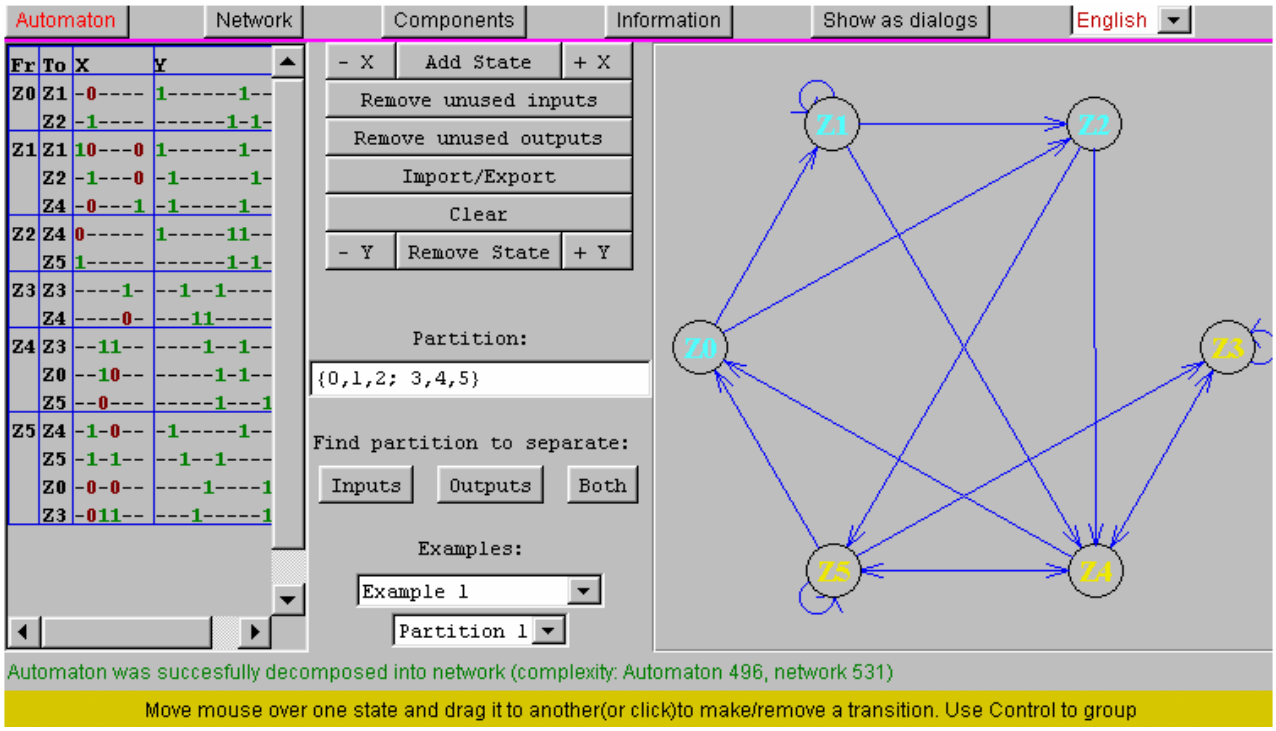

Fig.3. FSM decomposition applet

- *Microprogram table* is used to define the *control-path* of the system. During 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.

The teaching system is designed to operate in several working modes:

1) The *Design / Simulation* mode is used to define *data-path* (its internal structure is reflected in the *schematic view*) and *control-path* of the system. The control-path, which implements Mealy FSM, is defined in the *microprogram table*. This mode allows to run RT-level fault-free simulation, which can be executed for a single set of input data (step-by-step or at once) as well as for all the sequence of input operands at once. In the *step-by-step mode* each row of the microprogram is executed separately and results are constantly updated in schematic view panel. This mode is useful for illustration of the design work and for debugging.
2) In the *Test Mode* the applet is capable to perform fault simulation of the designed system. The fault simulation can be carried out at both functional and gate-level. The fault coverage information is provided both for the whole data path and for each single unit under test. In this mode gate-level schematics for a selected unit is displayed, which allows to perform manual local test patterns generation.
3) *BIST module* provides the basis to experiment with embedded self-test facilities. Two modes of BIST architectures are implemented: Logic BIST mode based on using random test pattern generator (TPG) and signature analyzer (SA), or Circular BIST mode based on using combined TPG/SA scan-path register [8]. 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 flexible design. The RT-level system model, shown in Fig. 1 is not mandatory. Should any other model be used, it must be only specified in a form of text-files. Then it can be loaded just as easily as the original one.

The applet has a built-in extendable collection of examples implementing different algorithms. They help users to understand the 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.

V. APPLETS FOR CONTROL PART DECOMPOSITION

The formal description of the control unit is a FSM which generates control signals to activate different operations in specific clock cycles. FSMs have been widely used also to express algorithms, communication protocols, digital systems, sequential logic circuits, and sequential logic cells.

This part of the learning system focuses on a specific but comprehensive problem of decomposition of FSMs. Decomposition has been a classic problem of discrete system theory for many years. FSM decomposition is a topic that waxes and wanes in importance. The fundamental works were done in the 1960s. During the

beginning of the VLSI era FSM decomposition became less interesting, and is becoming more important again with pervasive use of programmable logic and low power applications in digital design. 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.

Theoretical background of our system is the automata decomposition theory, which uses partition pair algebra proposed in [10]. The importance of this theory lies in the fact that it provides a direct link between algebraic relationships and physical realizations of finite state machines. The mathematical foundation of this theory rests on algebraization of the concept of "information" in a machine. It supplies the algebraic formalism necessary to study problems pertaining to the flow of this information in machines as they operate. As a part of engineering mathematics it falls squarely in the interdisciplinary area of applied algebra.

The problem of quality-driven synthesis corresponds to the optimal decomposition of a state machine reduced to a choice of partitions on the set of states of the prototype machine. Here, we are concerned with solving complex combinatorial tasks arising from the process of design.

In this part of the learning tool set, different applets for studying the basics of the decomposition theory of FSMs have been developed [7]. The applet on construction of a FSMs network (Fig. 3) allows experimenting with decomposition of the prototype machine. Different partitions can be chosen to decompose the given FSM to meet different design restrictions.

The developed set of applets can be used for teaching the basics of automata theory. Further basic applets in this area can be found at [11].

VI. CONCLUSION

The importance of teaching digital design and test is increasing with the growing complexity of electronic systems.

In this paper, we present a conception of how to improve the skills of the students studying digital design and test related topics. We apply a learning method based on using the so-called "living pictures". 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 use of the web-based media allows individual learning in accordance with the students' own needs. The principal message of the conception is to inspire students to learn, and to prepare them for developing problem-solving strategies.

VII. ACKNOWLEDGMENT

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 G4300).

VIII. REFERENCES

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

[2] M. Ercegovac, T. Lang, and J. H. Moreno, *Introduction to Digital Systems*, John Wiley & Sons, NJ: 1999, p. 498.

[3] J. E. Gilbert, *Arthur: an Intelligent Tutoring System with Adaptive Instruction*, Ph.D. thesis, University of Cincinnati, 2000, p. 75.

[4] R.Ubar, H.-D.Wuttke, "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.

[5] L. Anido, M. Llamas, and M.J. Fernandez, "Internet-based learning by doing," *IEEE Trans. on Education*, vol. 44, no. 2, May 2001, p. 18.

[6] R.M. Ford, J. Bondzie, and P. Kitcho, "Java applets for microelektronics education," *IEEE Trans. on Education*, vol. 44, no. 2, May 2001, p. 10.

[7] Teaching system URL: http://www.pld.ttu.ee/dildis/automata/applets

[8] H. Bleeker, P. van den Eijnden, F. de Jong, *Boundary-Scan Test: A Practical Approach*, Kluwer Academic Publishers, Dordrecht: 1993, p. 225.

[9] Texas Instruments' Scan Educator URL: http://www.ti.com/sc/data/jtag/scanedu.exe

[10] J. Hartmanis and R. E. Stearns. *Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, Engl. Cliffs, 1966, p. 209.

[11] Teaching system URL: http://kauai.theoinf.tu-ilmenau.de/forschung/projekte/sane/uebersicht_de.htm