DIGITAL DESIGN LEARNING SYSTEM BASED ON JAVA APPLETS

A. Jutman
PhD Student
15 Raja, 12618 Tallinn
Estonia
artur@pld.ttu.ee
http://www.pld.ttu.ee/

A. Sudnitson
Assoc. Professor
15 Raja, 12618 Tallinn
Estonia
alsu@cc.ttu.ee
http://www.pld.ttu.ee/

R. Ubar
Professor
15 Raja, 12618 Tallinn
Estonia
raiu@pld.ttu.ee
http://www.pld.ttu.ee/

ABSTRACT

In this paper, we offer tools which support the learning process in computer engineering area. The greater part of the tools is intended mainly to illustrate problems in control intensive digital systems such as investigation of tradeoffs 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. 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.

Keywords

Asynchronous-mode learning, Register-transfer level, Applets, Decomposition, Test

1. INTRODUCTION

To cope with today's demands in a very rapidly developing electronics industry, the engineering curricula and teaching technologies must be constantly updated. Entering into the System-on-Chip (SoC) era means that combination of theory and practice of digital design and test becomes an integral part of the VLSI and hardware computer architecture courses.

The paper presents a new teaching conception for distance-learning based on using of the so-called "living pictures". The field considered covers computer engineering specifically, switching and automata theories, and more, design and test of digital circuits and systems. A set of tools (interactive modules) is offered which support different stages of the learning process: class teaching, individual home training, and self-testing. On the one hand, teachers can demonstrate different examples and procedures of test related topics using computer simulated living pictures during their lessons. On the other hand, students can use the same simulations on their home computer, if the living pictures are available on the Internet. The core of the teaching concept presented are some JAVA-applets (interactive modules) running on any browser connected to the Internet. They can be accessed irrespective of time and place.

The paper is structured as follows. Section 2 describes the main teaching concept of the system under development. Section 3 is devoted to the register-transfer level design, simulation and test problems. Section 4 describes the applets developed for teaching some theoretical problems of finite state machine decomposition. Section 5 introduces an applet for teaching principles of boundary scan technique important for SOC design. Section 6 presents some conclusions.

4th Annual LTSN-ICS Conference, NUI Galway

2. BACKGROUND

The core of the teaching concept presented here is a Java-applet of a special type, which we call "Living Pictures" [8]. 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.

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).

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 [8]:

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.

In our teaching system [1] we succeeded to combine and illustrate many different problems related to control intensive digital design. This gives a unique possibility to teach all of them in a consecutive iterative approach.

3. 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 [1].

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 (Figure 1) consists of the following parts:

- *Schematic View* panel provides the schematic representation of the design and the graphical

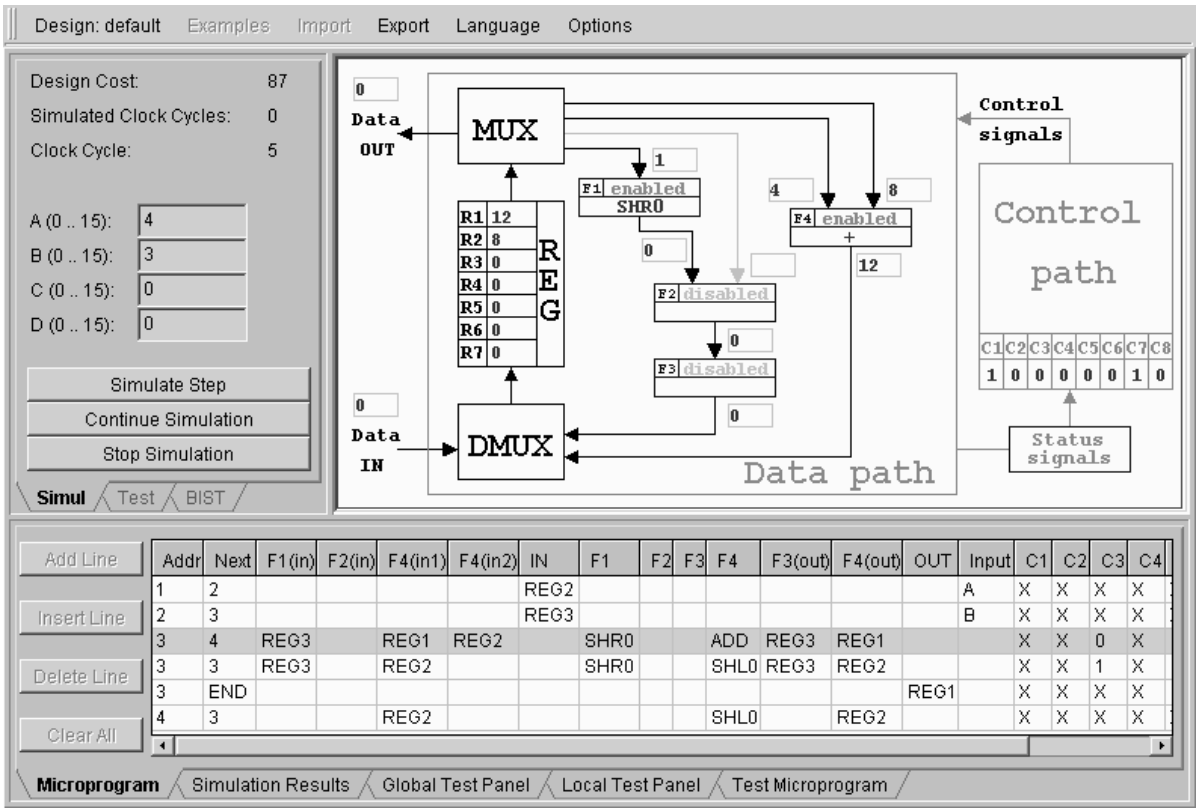

Figure 1. RT-level design applet window

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.

The simulation can be carried out in two different modes:

1) in the step-by-step mode or 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 one is testing the design repeatedly with some set of input data.

The applet has a flexible design. The RT-level system model, shown in Figure 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 simply as the original one. The Microprogram Table and the Simulation Results table will be automatically reconfigured as well.

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 [1]. 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.

4. CONTROLLER DECOMPOSITION

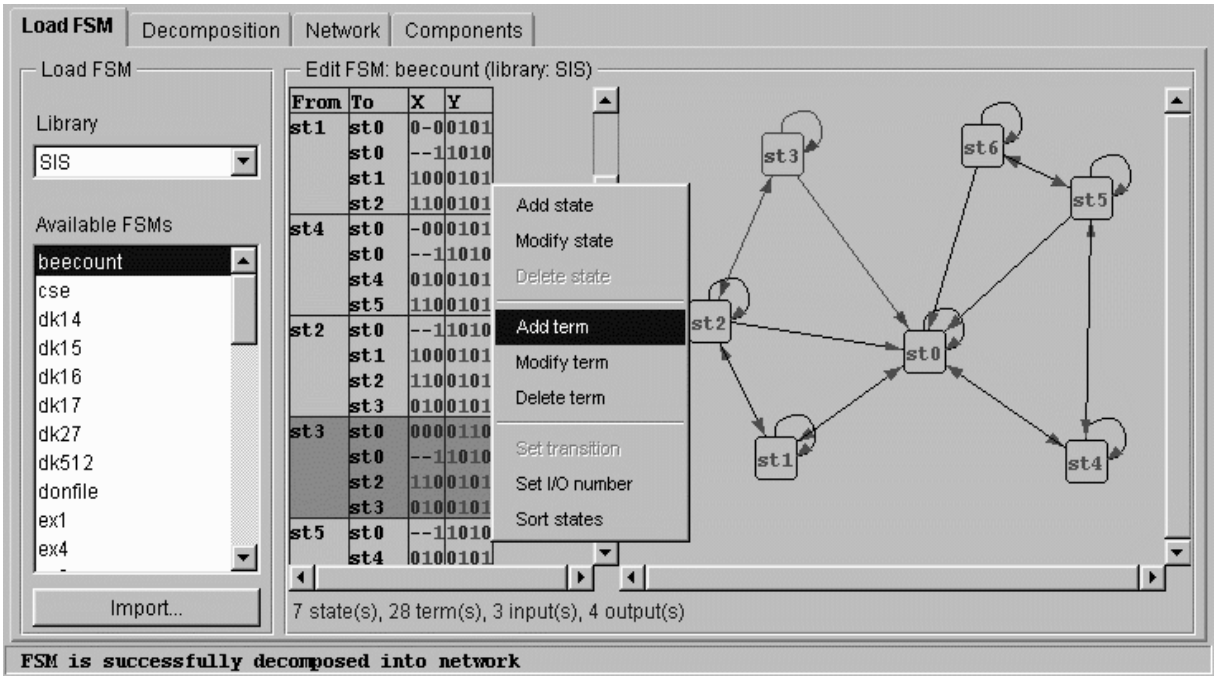

Figure 2. RT-level design applet window

The formal description of control unit is a Finite State Machine (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 learning system focuses on a particular but comprehensive problem of decomposition of FSM. The task of decomposition has been a classic problem of discrete system theory for many years. Decomposition of FSM is a topic that is becoming more important 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.

Theoretical background of our approach is the automata decomposition theory, which uses partition pair algebra proposed in [3]. 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 an algebraization of the concept of "information" in a machine and supplies the algebraic formalism necessary to study problems about the flow of this information in machines as they operate. It falls squarely in the interdisciplinary area of applied algebra, which is a part of engineering mathematics. We are concerned with solving complex combinatorial tasks arising from the process of design that is important for the student's theoretical background improvement.

In this part of the learning tool set, different applets for studying the basics of the decomposition theory of FSM have been developed [6]. The applet on construction of a network allows experimenting with decomposition of FSM. Different partitions can be chosen to decompose the given machine to meet different design restrictions.

The developed applet (Figure 2) 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 [5] during the lecture explaining the basics of the topic.

5. BOUNDARY SCAN LEARNING

Today's printed circuit boards (PCBs) have a number of inaccessible internal layers. 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.

A BS device manipulation is quite a tricky exercise. Therefore, only a system which allows instant simulation and illustration of all of 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, the Texas Instruments company has developed a training system called ScanEducator [6].

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 the 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 the Test Access Port (TAP), the TAP Controller and Boundary Scan Description Language (BSDL) [5]. All these structures do not seem that complicated when their operation can be dynamically illustrated using the concept of "Living Pictures".

The presented system demonstrates principles of testing chips, which have a BS structure inside. It provides that the user can experiment with a built-in collection of chips and boards, load those interesting for him directly from the Internet and combine his own boards. To better understand the main ideas of this standard our applet allows several working modes:

- Design/Editing of the BS structures inside the target chip using the BSDL language. In the Edit Chip 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 the 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 be interesting also for test engineers.

- Design/Description of the target board that consists of several chips. There are two possibilities to create boards: using interactive mode, adding new chips using dialog boxes and connecting/disconnecting them by clicking on their pins or using simple netlist language. The latter allows saving/loading the user's board descriptions from/into text files. The user can combine both methods, as he wants.
- Simulation of work of the TAP controller, scan register and other BS registers. 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.
- Insertion and diagnosis of interconnection faults. 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.

6. CONCLUSION

The importance of test and design for testability problems grows along with the complexity of electronic systems, since the expenses of verification and testing are becoming the major components of the design and manufacturing costs of new electronic products.

In this paper, a conception of how to improve the skills of students studying digital design and test-related topics is presented. We suggest a learning method based on using the "living pictures". The method presented here deals with the goal to put interactive teaching modules to the Internet so that they can be used by a lecturer as well as for individual self-studies.

The use of the web-based media makes it possible to achieve: independent of place and time access to the course material, individual learning in accordance with the student's own needs, new forms of communication between the teacher and student.

The conception presented allows improving the skills of students studying electronics design and test-related topics. The principal mission of the conception is to inspire students to learn, and to prepare them for developing problem-solving strategies.

7. 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 G4300).

8. REFERENCES

[1] Abramovici M., *Digital Systems Testing and Testable Design*. IEEE Press (1999).

[2] Digital Design and Test Teaching Web Site: http://www.pld.ttu.ee/dildis/automata/applets

[3] Hartmanis J., and Stearns R.E., *Algebraic Structure Theory of Sequential Machines*. Prentice-Hall, Engl. Cliffs (1966).

[4] Kenneth P. Parker, *The Boundary-Scan Handbook.* Kluwer Academic Publishers (1999).

[5] McElvain, K.: LGSynth'93 Benchmark Set: Version 4.0, 1993. Available: http://www.cbl.ncsu.edu/benchmarks.

[6] Teaching System of Digital Design URL: http://www.pld.ttu.ee/dildis/automata/applets

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

[8] 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*, 107-110 (2000).