

Self-Learning Tool for Digital Test

Raimund Ubar, Elmet Orasson, Teet Evartson

Tallinn Technical University

Raja 15, 12618 Tallinn, Estonia

Tel. 372 620 2252; Fax: 372 620 2253; {raiub,elmet,teet}@pld.ttu.ee.

Abstract

The paper presents a new teaching concept based on using "living pictures" supporting the learning process by the possibility of distance learning as well as a web-based computer-aided teaching. A set of tools ("interactive modules") is offered to support this concept in teaching digital test and diagnosis related topics in the university courses of computer engineering, switching and automata theories. A big reservoir of examples and the possibility to generate own ones makes the learning process more interesting and allows learning at an individual depth and duration. The interactive modules are focused on correct solutions, easy action and reaction, multilingual descriptions, learning by doing, a game-like use, and fostering students in critical thinking, problem solving skills and creativity.

1. Introduction

The more complex are getting electronics systems, the more important will be the problems of test and design for testability because of the very high cost of testing electronic products. At present, most VLSI and system designers know little about testing, so that 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 [1]. This reflects also today's university education: everyone learns about design, but only truly dedicated students learn test. The next generation of engineers involved with VLSI and System-on-Chip (SoC) technology should be made better aware of the importance of test, and trained in test technology to enable them to produce high quality, defect-free products.

In this paper a conception is presented how to improve the skills of students to be educated for hardware and SOC design in test related topics. We present a learning method based on using so-called *living pictures* [2,3]. The method presented deals with the goal, to put interactive teaching modules to the Internet that can be used in a lecture as well as for individual self-studies [2]. They can be accessed independent of time and place. On 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 (the interactive modules) running on any browser connected to the Internet. We call this type of applet "Living Pictures". By using interaction possibilities the students can generate examples that are

interesting enough to encourage their own experiments. They can produce input stimuli and watch the behaviour of the circuit in the fault-free mode and also in different faulty modes.

2. Description of the user interfaces

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:

- Vector insertion panel
- View panel for design schematics
- View panel for displaying information

The vector insertion panel has two subpanels for inserting single input test vectors and for setting up the feedback configuration of a Linear Feedback Shift Register (LFSR) to be used for automatically generating test vectors. In the LFSR mode, the first subpanel is used for initializing the LFSR. The LFSR based Automated Test Pattern Generator (ATPG) is used for emulating different BIST ideas like Built-In-Logic-Block-Observer (BILBO) or Circular-Self-Test-Path (CSTP) [4,5]. The first subpanel is also used when creating test vectors for specific fault detection. In this case, the fault activating and propagating values are inserted one by one into the signal boxes at connections of the design schematics, and the input test vector will be deduced from these internal signal values.

The schematics panel displays currently selected schematics. The small boxes at the lines display internal signal values on connections. The boxes are clickable 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.

Detected faults, signal conflicts etc. are displayed as colored bold wires. Color coding is as following: red - stuck-at-1 fault is detectable, green - stuck-at-0 fault is detectable, gray - undefined (don't care) signal, and blue - conflicting signals.

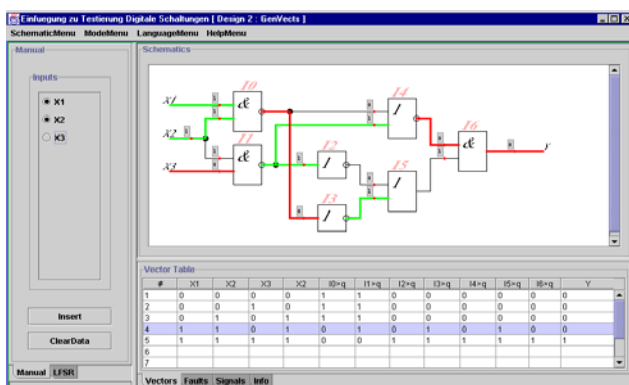


Figure 1. “Living picture” as Java applet

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

3. Main functionalities of the applet

There are four main menus used with the applet: schematics, mode, language, and help. The 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 possible modes: sequential and combinational diagnosis).

We start working with the applet by selecting a circuit from a set of predefined circuits. Then we can carry out different experiments with this circuit by selecting a proper working mode from the mode menu.

In the vector insertion mode we choose test vectors either automatically by using LFSR, or by inserting vectors manually. In the manual mode, we generate step by step input patterns which are simultaneously simulated. The boxes at the lines on the schematics subpanel display the result of simulation – the values of internal signals on the connections. The waveforms can be viewed on the data subpanel .

When using LFSR, we have to specify the initial state, to set up the feedback structure, and to specify the length of the test sequence. By LFSR we can simulate the BIST either in the mode of Built-In Logic Block Observer (BILBO) or in the mode of Circular Self-Test Path (CSTP) [4,5]. By changing the settings on the vector insertion panel we can emulate different feedback structures of the chosen BIST architecture.

In the test generation mode we choose a target fault in the schematic and create step by step proper activated paths in the circuit to activate the fault at his site and to propagate the error signals caused by the fault towards output by clicking the needed values into boxes on the lines. From these values finally, an input vector will be deduced. The colours on lines help us 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 colour. As the result of the procedure, a test pattern will be generated. The detected by the test faults are displayed also on the data panel in form of a row in the fault table.

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

In the fault diagnosis mode we need at first, to create a fault table by running the fault simulator for a set of previously generated test vectors. Entering into the diagnosis mode will insert a random fault into the circuit. The following diagnosis strategies chosen from menu can be investigated: combinational and sequential diagnosis. For learning the combinational diagnostic strategy, a single vector or a subset of vectors can be selected and applied to the erroneous circuit (by imitating test experiments). The applet shows the results of testing, and displays also the subset of suspected faults. 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 the both diagnostic strategies is to try to localize the fault 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.

As an example in Figure, a test generation mode is activated for a design No 2. The result is shown on the schematic panel in the form of activated paths and detected faults. The values in boxes show the behaviour of connection lines of the circuit for the test vector No 4. The activated faults are highlighted by coloured lines, the value 0 (or 1) in the boxes means that the fault stuck-at-1 (or stuck-at-0) is activated. Only a single path from the input X3 through the gates I1, I4 and I6 is activated up to the output Y, hence, only the faults on that path can be detected and observed on the output. The detected faults are: stuck-at-0 on the gate output I1, and stuck-at-1 on the input X3 and on the gate outputs I4 and I6. There are other faults also highlighted and activated inside the circuit on the inputs X1, and X2, and on the gate outputs I0 and I3, however, these faults are not detected because the activated path is broken on the gate output I5, and the error signal is not propagating up to the observable output Y. The data panel shows 5 simulated test patterns, the pattern No 4 which was analysed above is selected.

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 basics of the topic. The applet can be used also 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.

4. Conclusion

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, quick cross-referencing by hyper-linked texts, new forms of communication between teachers and students (chat, joint editing), 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. The principal mission of the conception is to inspire students to learn, to inspire them on a journey to knowledge, and to prepare them to develop problem-solving strategies.

5. Acknowledgments

This work was supported by the Thuringien Ministry of Science, Research and Art (Germany), by EU Copernicus project VILAB, and by the Estonian Science Foundation, grants No 3658 and 4300. We thank also Dr. Wuttke from TU Ilmenau, Germany for close cooperation in developing this applet.

References

- [1] O. Jones, “Article Title”, *Journal*, Volume, Publisher, Location, Date, pp. XX-YY.
- [2] M.L. Bushnell. Increasing Test Coverage in a VLSI Design Course. International Test Conference, Atlantic City, NJ, USA, 1999, p. 1133.

- [3] 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, May 6-8 1999, pp. 50-54.
- [4] 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.
- [5] M.Abramovici et al. *Digital Systems Testing and Testable Design*. IEEE Press, 1999, 652 p.
- [6] M.L. Bushnell, V.D. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000, 690 pp.
- [7] <http://www.pld.ttu.ee/tt/>