WEB-BASED APPLET FOR TEACHING BOUNDARY SCAN STANDARD IEEE 1149.1

A. JUTMAN, A. SUDNITSON, R. UBAR
TALLINN TECHNICAL UNIVERSITY, ESTONIA

KEYWORDS: Web-Based Teaching, Boundary Scan, Java Applet

ABSTRACT: A conception of training system for teaching the IEEE 1149.1 Boundary Scan (BS) standard is presented. The system conforms to the idea of “Living Pictures” [2] where a tricky quite complicated content is represented in a graphical way and the student is provided with the possibility to “play” with the training system. The design/modification of BS structures inside the target chip; design/modification of the target board consisting of several chips; simulation of work of TAP controller, scan register and other BS registers; investigation of BS commands like EXTEST, BYPASS, etc.; insertion and diagnosis of interconnection faults – all these topics are supported by our teaching system. The system is implemented in a form of Java applet and can be freely accessed over Internet. The latter makes it easy for students even from foreign universities to use this system any time and in any place.

INTRODUCTION

Rapid advances in areas of deep-submicron electron technology and design automation tools enabled engineers to design larger, more complex, integrated circuits. Until recently, most electronic systems consisted of one or multiple printed circuit boards (PCB), containing multiple integrated circuits (IC) each. Recent advances in IC design methods and technologies allow integration of these complex systems onto one single IC. These developments are driving engineers toward new System on a Chip (SoC) design methodologies. SoC is seen as a major new technology and the future direction for the semiconductor industry.

The importance of test and design for testability problems grows together 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. The main test strategy for SoC is the IEEE P1500 “Standard for Embedded Core Test.” This standard has evolved from the earlier development in this area, namely, the IEEE Std 1149.1 “Test Access Port and Boundary-Scan Architecture”, which was created by Joint Test Action Group (JTAG) and aimed at testing PCBs. It is of no doubt that these new very important standards have to be taught to future designers and test engineers. This holds in the first place for the latter and the simpler one, shortly called the Boundary Scan (BS) standard.

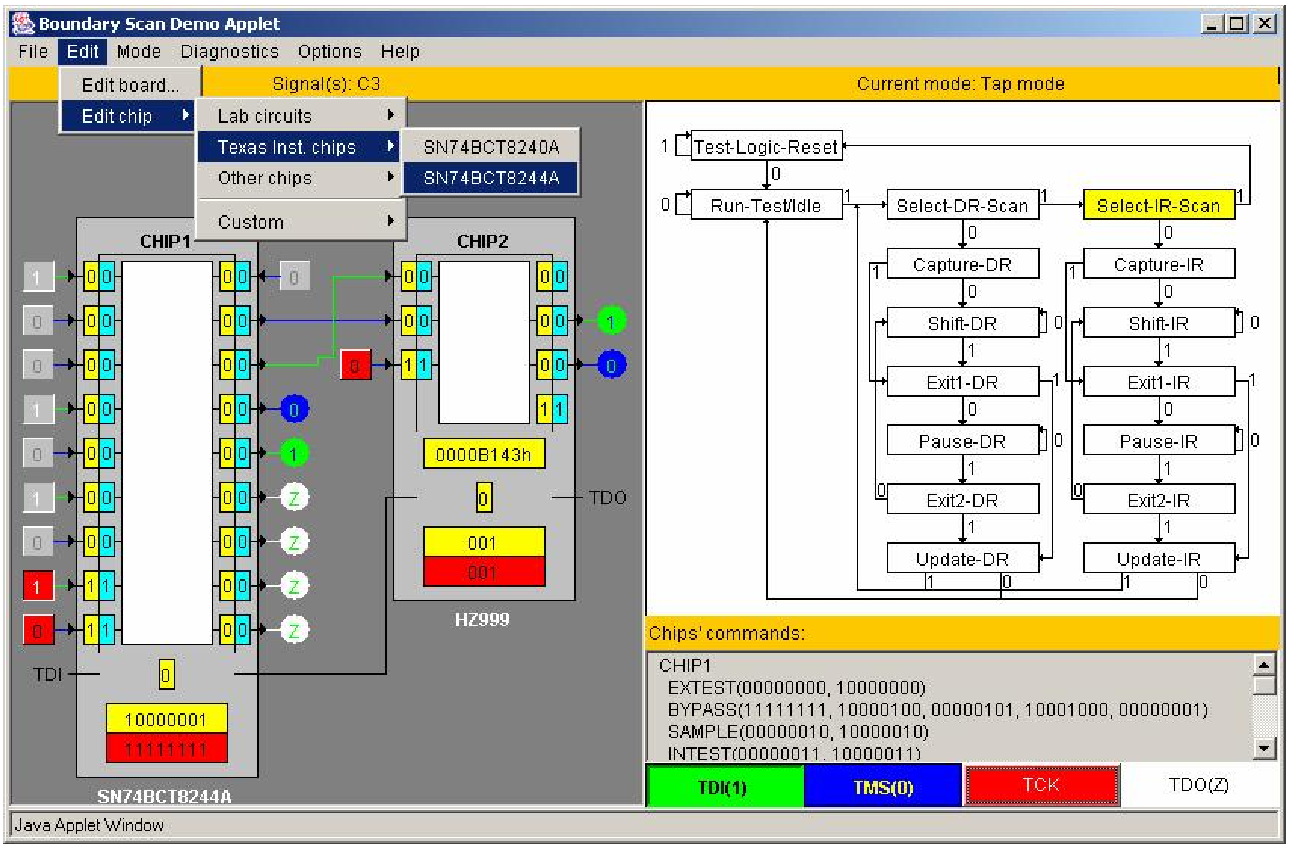
Before the BS standard was introduced, in 70s and 80s the main board test technique was the “bed of nails”, which ensured the physical contact to any desired point of the circuitry. This method, however, was limited to boards with two conductive layers only. Moreover, the circuit packaging was restricted to the “dual in line”. Modern PCBs have a number of inaccessible internal layers as a rule, and the modern circuit packaging includes types like various “grid arrays” (PGA, BGA), which have a lot of inaccessible pins. These difficulties have led to introduction of the Boundary Scan standard.

The BS architecture implies the insertion of scan chains in such a way, that each pin of each chip receives an internal control point. During the normal operation mode the scan cells are transparent, while in the test mode they provide with possibility of driving any of the control points to a desired value as well as reading the results of circuit operation. The standard defines also the Test Access Port (TAP) and the TAP Controller [1]. All these structures are quite complicated and hard to be taught in a traditional manner during a lecture. Therefore, new effective dynamic teaching concepts must be introduced.

We apply the concept of “Living Pictures” [2] when building up our teaching system. The main features of such a system incorporate: graphical representation of the learning subject, dynamic content, user-friendly interface, concentration on the most important topics in the simplest possible way, set of various examples, easy action and reaction, and game-like style of learning.

The same system could be used by teacher during a lecture when explaining the dynamic content as well as by students later at home when repeating and comprehending the topic. In this way this dynamic part of the lecture will not be lost. Moreover, the same system could be used during tests or examinations. Therefore, it must be easily available. How to keep up with all these requirements? The solution is to implement the system on the Java platform, which is supported by most operating systems, and put it into the Web.

In the next section we give a general overview of our system and compare it with a similar system called ScanEducator from Texas Instruments [3]. The description of the Schematic View Panel together with BS registers comes in Section 3. In the section after, we



Signal(s): C3

Current mode: Tap mode

Chips' commands:

```
CHIP1
  EXTEST(00000000, 10000000)
  BYPASS(11111111, 10000100, 00000101, 10001000, 00000001)
  SAMPLE(00000010, 10000010)
  INTEST(00000011, 10000011)
```

Fig. 1 Applet main window

describe the chip and board editing capabilities of the system. The overview of both TAP and Command working modes is given in Section 5. A scenario of a possible practical work on basics of BS is described in Section 6. Final sections of the article bring some conclusions and acknowledgements.

SYSTEM OVERVIEW

We have selected the Java technology for our teaching system since it is well supported by main operating systems like Windows, Linux, and Solaris. Furthermore, Java allows for graphical content to be easily created. It also has well developed means for creating the user interface. We made our system available in the Web for free [4].

All the mentioned advantages of the Java environment are at the same time the advantages of our teaching system against the ScanEducator [3] from Texas Instruments, 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 number of built-in examples and a flexible system of example management. The users can generate their own examples by creating a fully custom chip or board. Moreover, our applet has a failure insertion and fault diagnosis possibility.

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

- simulation of operation of TAP Controller;
- illustration of work of BS registers;
- insertion and diagnosis of interconnection faults;
- design/editing of BS structures inside the target chip using the BSDL language [1];
- design/description of a target board that consists of several chips.

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 BSDL format which is a part of the standard. 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 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.

At the moment, the only supported description of the chip's internal logic is the SSBDD format [5]. However, this format is optional since for the simulation of the most BS operation modes the internal structure of the chip can be neglected.

The simulation of chip operation 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 of BS commands like EXTEST, SAMPLE/PRELOAD, etc. with different predefined input data.

The same mode is useful 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.

BOARD REPRESENTATION

The Schematic View Panel (Fig. 2) comes with the schematic representation of the board layout. At the moment, possible elements of the board are restricted to chips themselves only. This means that only components which have BS structures can be represented and simulated in the applet. The BS standard distinguishes the following basic hardware elements: the *Test Access Port*, the *TAP Controller*, the *Instruction Register*, and a group of *Test Data Registers*, which in its turn consists of at least the *Boundary Scan Register* and the *Bypass Register*. All these structures are also illustrated in the Schematic View Panel.

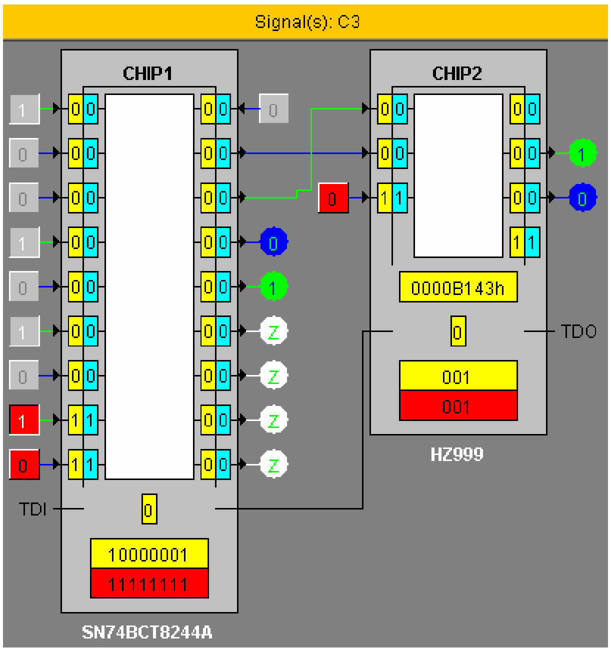


Fig. 2 Schematic View Panel

Illustration of a BS Chip

A BS chip is shown in the Schematic View Panel as a grey rectangle. According to the BS standard all the chips on the board are connected into a scan chain via TDI (Test Data In) and TDO (Test Data Out) pins.

The white area inside the chip is the core logic, which is not illustrated in detail. The wrapper around the core logic is the Boundary Scan Register. It plays the role of separator between the logic circuitry and the I/O pins of the chip, providing internal control points for each pin. Each scan cell of this register consists of two flip-flops. One is used for capturing the state of the control point (the yellow one). Another one (the blue one) is needed for keeping the control value of the control point during the test mode. The control values and captured responses are shifted in and out in series via TDI and TDO pins.

The register at the very bottom of the chip is the Instruction Register (IR), the lower part (red) of which is used for holding the currently active instruction. The next instruction is shifted in via the upper part of IR. After the instruction has been completely shifted in, the lower part is updated with the values from the upper part.

The one-bit register right above the IR is the Bypass Register (BYR). It is usually used for faster test data shifting. The IR and the BYR are mandatory Test Data Registers (TDRs).

There could be some optional design-specific TDRs in the BSDL description that are not normally shown upon the chip in the Schematic View Panel. The only *optional* TDR that is shown accordingly to the BSDL description is called the Device ID Register.

There is a library of different chips and a number of built-in examples of various board schematics provided with our applet. So, the lecturer can select the most illustrative example for a particular topic.

Signals and Wires

We use different colors for input pins as well. Ordinary data input pins are grey, while the control pins are red. By pushing the pins (the buttons) the user can drive a desired input line to either logic 0 or 1. Depending on the BSDL description for a particular chip the value 0 or 1 at a control input pin drives some of the outputs to the high impedance state.

Each output which is not connected to any other chip, is supplied by an indicator, which shows the current state of the output line. There are four possible states for any signal line of the design. Logic 1 and logic 0 are indicated by green and blue colors correspondingly. The wires driven into the *high impedance state* are white. At the same time the output indicator holds value “Z”. If a wire is red and the indicator shows value “X”, then the logic value of the corresponding signal is *unknown*.

The wire connecting all the TDIs and TDOs of all the chips together is always black. It is used for test data shifting in and out.

Different colors highlighting different things are selected in the way making it easy to follow the simulation. Any change, when it happens can be quickly noticed.

BOARD AND CHIP EDITING

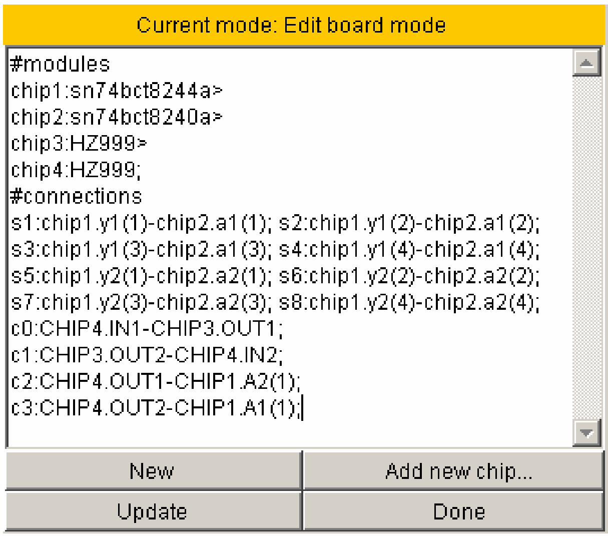
Our applet has a built-in BSDL parser which makes it possible for virtually any BS chip to be imported and simulated by our system. Moreover, the built-in simple board editor allows for any design to be created using arbitrary BS chips. We decided to do so, because it makes the work with the applet more exciting for a student. Even a test engineer may find some of the applet's functionality useful for certain simpler tasks.

The user can select either one of built-in examples to be modified or he can start with an empty design to build everything from scratch.

The BSDL language is the standard for BS structures description. It provides the applet with information sufficient for simulation of most of the BS instructions. However, this format does not describe the internal functionality of the chip. Therefore, we have to use another format for this purpose. We have selected SSBDD (Structurally Synthesized Binary Decision Diagram) format [5] for its simplicity and efficiency. This format can be easily converted from a widely-used EDIF description by Turbo Tester software [6].

During the definition of a new chip, the user should provide both formats for complete description of the chip. However the SSBDD format may be omitted if there is no need to simulate the chip's internal logic.

We use our own very simple format for description of the interconnection between the chips. It consists of two parts: description of modules (chips) and description of connections (Fig. 3). In the Edit Board mode, the user can edit both parts by adding or removing some lines. The chips in "modules" section should come in the same order as they will be connected into the BS chain. They are separated by the sign ">", which symbolically illustrates the ordering relation.



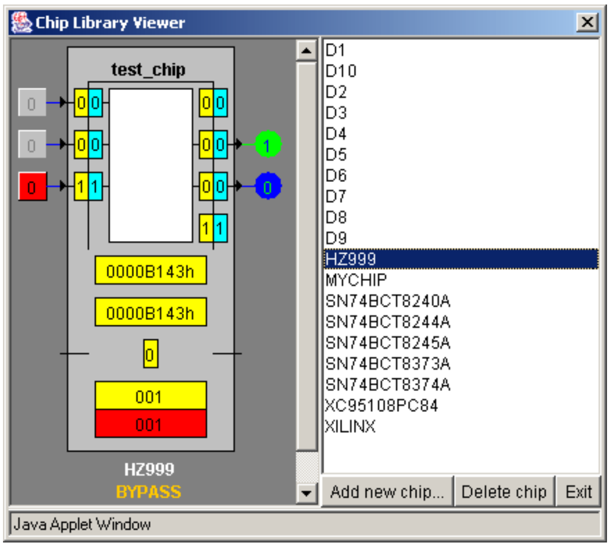
Current mode: Edit board mode

```
#modules
chip1:sn74bct8244a>
chip2:sn74bct8240a>
chip3:HZ999>
chip4:HZ999;
#connections
s1:chip1.y1(1)-chip2.a1(1); s2:chip1.y1(2)-chip2.a1(2);
s3:chip1.y1(3)-chip2.a1(3); s4:chip1.y1(4)-chip2.a1(4);
s5:chip1.y2(1)-chip2.a2(1); s6:chip1.y2(2)-chip2.a2(2);
s7:chip1.y2(3)-chip2.a2(3); s8:chip1.y2(4)-chip2.a2(4);
c0:CHIP4.IN1-CHIP3.OUT1;
c1:CHIP3.OUT2-CHIP4.IN2;
c2:CHIP4.OUT1-CHIP1.A2(1);
c3:CHIP4.OUT2-CHIP1.A1(1);
```

Fig. 3 Board Interconnect Description

Connections between chips can be created also by clicking upon corresponding pins with mouse pointer. Unneeded connections can be deleted in the same way. All the changes will be instantly reflected in the "connections" section of the interconnect description.

There is a convenient Chip Library Viewer (Fig. 4), which can be used during the board design for better overview of available BS chips. It provides a good access to all the chips in the library. The user can simulate separate chips, delete, or edit them. The user can also add new chips to the library using this panel.



D1
D10
D2
D3
D4
D5
D6
D7
D8
D9
HZ999
MYCHIP
SN74BCT8240A
SN74BCT8244A
SN74BCT8245A
SN74BCT8373A
SN74BCT8374A
XC95108PC84
XILINX

Fig. 4 Chip Library Viewer

SIMULATION AND FAULT DIAGNOSIS

There are basically two main simulation modes supported by the applet. They are: the TAP Controller mode and the Command mode. In both modes there are two options: fault-free simulation and simulation for fault detection. The latter option implies that there is a certain faulty connection. Then the goal of simulation is to find that faulty connection and identify the type of fault. This is done by selection of proper BS instructions and input test data. There are basically two possible types of faults available up to now: stuck-at 0 and stuck-at 1. We are also going to implement other faults like shorts between adjacent lines or broken connections.

The TAP Controller Mode

The TAP Controller simulation mode is illustrated in Fig. 1. In this mode, the user himself should take care of shifting in needed instructions and test data. Since, the Test Access Port consists of only TDI, TDO, TMS (Test Mode Select) and TCK (Test Clock) I/O pins, the task of the student becomes not quite trivial. All the instructions and test data can be shifted in via the TDI input only. The TAP Controller state together with an active instruction in the IR defines the operation and configuration of BS structures. For example, in order to select the EXTEST instruction, one should, first, reach the "Shift-IR" state of the TAP Controller starting from the "Test-Logic-Reset" and moving through the following states: "Run-Test/Idle", "Select-DR-Scan", "Select-IR-Scan", etc. These state transitions are made by changing the TMS value and applying the TCK.

The second phase is to shift in a proper bit sequence, which corresponds to the EXTEST instruction. For example, it is either 00000000 or 10000000 for SN74BCT8244A. This information can be found from the Chip's Commands sub-panel (see Fig.1).

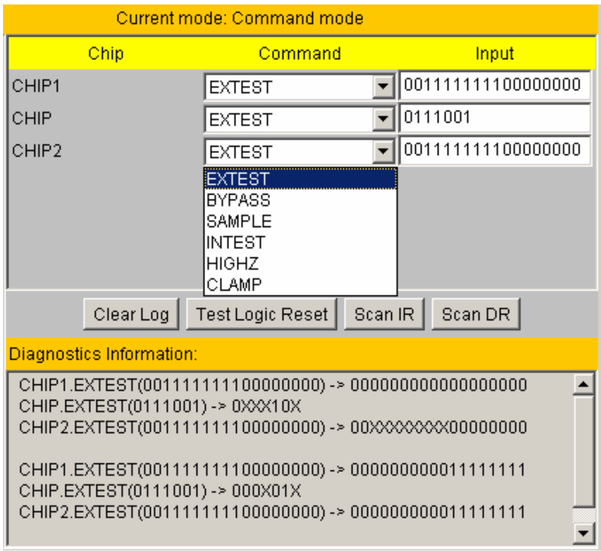
After that, when the chip is in the EXTEST state, the test data should be inserted into the Boundary Scan Register. This is done in the same manner, but in the “Shift-DR” state.

As one can see the manipulation of a BS chip is quite a tricky task. Therefore, only a system, which allows instant simulation and illustration of all the student’s steps can help learning and easy finding all the mistakes and misunderstandings, which otherwise would likely be missed.

The Command Mode

Students, once, become familiar with the TAP Controller operating modes and learn how different BS instructions work. They may want then to perform some time consuming operations automatically while concentrating on other aspects and applications of BS like, for instance, fault diagnosis.

During fault diagnosis it is convenient to set up, at first, special operating modes for each chip and then to concentrate primarily on test vectors selection in order to detect and locate faulty lines. This is possible in a special BS simulation mode of our applet called the Command Mode.



Current mode: Command mode

Chip	Command	Input
CHIP1	EXTEST	001111111100000000
CHIP	EXTEST	0111001
CHIP2	EXTEST	001111111100000000

Diagnostics Information:

```
CHIP1.EXTEST(001111111100000000) -> 000000000000000000
CHIP.EXTEST(0111001) -> 0XXX10X
CHIP2.EXTEST(001111111100000000) -> 00XXXXXXXX00000000

CHIP1.EXTEST(001111111100000000) -> 000000000011111111
CHIP.EXTEST(0111001) -> 000X01X
CHIP2.EXTEST(001111111100000000) -> 000000000011111111
```

Fig. 5 Instruction Selection Panel

There is a list of available instructions for each chip. Those instructions are defined in the BSDL description and could be different for different chips. The user just selects its own instruction for each chip and defines the input data (Fig. 5). Usually, depending on the active instruction, the input data will be shifted into the Boundary Scan or Bypass Register. By pressing “Scan IR” button the user makes selected instructions active. He can press then “Scan DR” button and the input data will be shifted in. At the same time the current state of Data Registers will be shifted out. It is a bit sequence which is shown in the Diagnostic Information sub-panel.

For a certain test vector there is an expected output response. When the response is different from what was expected, the fault has been detected. It is possible to locate the faulty connection by analyzing the data in the Diagnostic Information sub-panel.

In the Command mode there are two possible simulation options: slow animation mode and fast mode. The first one is intended for comprehensive illustration of various processes like shifting or switching taking place in the chip and different BS registers. The second mode is used for fast simulation without animation. It is useful when the primary goal is to get the diagnostic information.

UNDERSTANDING THE BS: A PRACTICAL WORK SCENARIO

In this section we introduce a possible scenario of a practical work on the basic concepts of Boundary Scan supported by the BS applet. It is mostly meant for very beginners - students or engineers. They should learn the BS instructions and working modes and see, from inside, how the BS structures are operating. First of all, students should study the TAP Controller, which is the key device in the whole BS conception. They learn to move from state to state on the state diagram and insert different BS instructions via TDI. The next step is the study of data registers and their proper use.

When the main principles of BS operation are understood, the students face the task of interconnect diagnosis. They should learn how to properly select test vectors in order to find interconnect defects of a given type. The final and the most advanced task is to write a description of an own chip according to given parameters using BSDL format. The practical work scenario consists of 6 steps and is given in the following.

Scenario.

1) Understanding the TAP controller. Using TMS signal and the test clock, students should move through the state diagram of the TAP controller starting from Test-Logic-Reset state towards some given state and return back. They should find the answer to the question: what is the maximum number of clock cycles needed to return to Test-Logic-Reset state from any random state if the TMS kept constantly 1.
2) Studying the Instruction Register. Finding the IR branch on the TAP controller state diagram and practicing with capturing and shifting in given instructions using TDI signal. Working with several chips simultaneously and learning shifting different instructions into different chips in a correct way.
3) Learning the Boundary Scan instructions and the working modes of Boundary Scan. Understanding EXTEST, INTEST, BYPASS, IDCODE, SAMPLE, CLAMP, and HIGHZ instructions by setting various chips into different modes and simulation of their operation.

4) Using the Data Registers. Reading chip's ID from IDCODE register. Bypass register and BYPASS instruction. Selection of the BS register by EXTEST and INTEST instructions. Shifting different given test vectors into several connected chips simultaneously.
5) EXTEST instruction and interconnect diagnosis. Preparation for interconnect diagnosis by setting up all the chips in EXTEST mode. Fault insertion and its diagnosis by selection of proper test vectors.
6) Development of a custom chip description using the BSDL language (advanced exercise). Chip verification and simulation.

CONCLUSIONS

In this article, we have described a teaching system [4], which is developed in order to make learning of the IEEE Std 1149.1 "Test Access Port and Boundary-Scan Architecture" standard an easy task.

A 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.

The system consists of several interactive modules and supports the possibility of distance learning as well as a web-based computer-aided teaching. The interactive modules are focused on easy action and reaction, learning by doing, a game-like use, and encourage students for critical thinking, problem solving, and creativity.

We presented also a scenario of a possible practical work on basics of BS, which is supported by our teaching system. The work is meant for the beginners. It can be used as by teacher during laboratory exercises as by students for independent self study.

THE AUTHORS

A. Jutman, A. Sudnitson and R. Ubar are with the Computer Engineering Department of Tallinn Technical University, Raja 15, 12618 Tallinn, ESTONIA. E-mail: artur@pld.ttu.ee

ACKNOWLEDGEMENTS

This work is supported by the Thuringian Ministry of Science, Research and Art (Germany), by EU V Framework project REASON, and by the Estonian Science Foundation (grants No 5643 and 4300). We also thank Ruslan Gorjachev and Vjatseslav Rosin for the perfect implementation of our ideas in the form of Java applet and Dieter Wuttke for valuable comments and hospitality.

REFERENCES

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

[2] H.-D. Wuttke, K. Henke, "Teaching Digital Design with Tool-Oriented Learning Modules "Living Pictures", in *Proc. of 32nd ASEE/IEEE Frontiers in Education Conference*, Nov. 6 - 9, 2002, Boston, USA, Session S4G, pp. 25-30.

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

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

[5] A. Jutman, J. Raik, R. Ubar, "SSBDDs: Advantageous Model and Efficient Algorithms for Digital Circuit Modeling, Simulation & Test," in *Proc. of 5th International Workshop on Boolean Problems*, Freiberg, Germany, Sept. 19-20, 2002, pp. 157-166.

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