www.czasopisma.pan.pl PAN www.journals.pan.pl
POLSKA AKADEMIA NAUK
jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2012, VOL. 58, NO. 3, PP. 233–239
Manuscript received February 20, 2012; revised March, 2012. DOI: 10.2478/v10177-012-0032-4

Open-Source JTAG Simulator Bundle for Labs

Konstantin Shibin, Sergei Devadze, Vjatseslav Rosin, Artur Jutman, and Raimund Ubar

***Abstract*—This paper presents a software/hardware bundle for studying, training and research related to IEEE 1149.1 Boundary Scan (BS) standard. The presented package includes a software environment Trainer 1149 that is capable to graphically visualize BS facilities and perform fine-grain simulation of BS test process. Trainer 1149 provides a cozy graphical design and simulation environment of BS-enabled chips and non-BS clusters. It provides the user with a full flexibility in working with any type of BS structures by supporting standard formats such as Boundary Scan Description Language and SVF (for defining test patterns). A special fault simulation mode allows injecting various types of interconnection faults to simulate their impact and inspect them using interactive tools. Trainer 1149 is the main component of a recent goJTAG initiative that aims at bringing JTAG tools closer to the user for both learning and experimental work purposes. The software part is implemented in multi-platform Java environment and distributed as an open-source freeware. Using a convenient low-cost USB-JTAG controller, one can also test real defects in real hardware. Such combination of features is unique for a public domain BS package.**

***Keywords*—JTAG, boundary scan, IEEE 1149.1, Trainer 1149, goJTAG.**

I. Introduction

As long as Printed Circuit Boards (PCB) will continue to exist, the PCB testing will remain a very important step in the production cycle of microelectronic systems. It has already become a mature research and engineering topic with well established standards and solutions the most important of which is the IEEE Std 1149.1 “Test Access Port and Boundary-Scan Architecture”, developed by Joint Test Action Group (JTAG) and balloted as a standard in 1990 [1]. The state of the art PCB testing is a mixture of Boundary Scan (BS), optical/x-ray inspection, and in-circuit test with the latter seizing to exist.

The BS has proven to be the most universal and the only realistic low-cost solution, which besides manufacturing testing is used also for in-circuit programming and product maintenance.

Advanced courses in microelectronics education must follow the latest industrial and research trends in order to supply the society with high-level engineers and researchers. Accordingly to the International Technology Roadmap for Semiconductors (ITRS) [2], the manufacturing testing of semiconductors and microelectronics has always been an area of special concern, which importance is even more increasing now due to new extremely complicated design techniques and advanced technologies. The current paper addresses this concern by introducing a training/learning package in the field of microelectronics reliability.

According to the European University Association, the learning process must change from teacher-centric to student-centric concept that will enable students to become the engaged subjects of their own learning process. It should also contribute to improving many issues of progression between learning cycles, institutions, labor markets, and countries [3]. The training tool we present in this paper is aimed at facilitating these goals by providing an environment where the student can deeply immerse himself into the studied subject by exploring numerous concepts that are implemented in a great detail. Unlike commercial/industrial software is build up based on fully automated push-button concept, Trainer 1149 [4] allows students to follow the basic and advanced principles behind complex algorithms it illustrates.

The tool is adapted for both *analytic* and *synthetic* study, where the students first learn the subject by observation (using prepared examples) and then generate and/or solve their own specific exercises. Trainer 1149 was designed with accordance to the concept of “Living Pictures” [5]. The main elements of this concept incorporate: graphical representation of the learning subject, dynamic content, user-friendly interface, concentration on the most important topics in the simplest possible way, easy action and reaction, and game-like style of learning.

The same system could be used by teacher during a lecture for explaining a dynamic content as well as by students later at home when repeating and digesting the topic. In this way the dynamic part of the lecture will not be lost. Moreover, the same system could be used later – during tests and examinations.

The current paper describes a multi-functional software system, developed in the first place for demonstration and simulation of different aspects of BS concept but also as a CAD environment for training, research, and development related to IEEE 1149.1 standard. We describe usage scenarios and show advantages of our system over two other similar public domain JTAG tools.

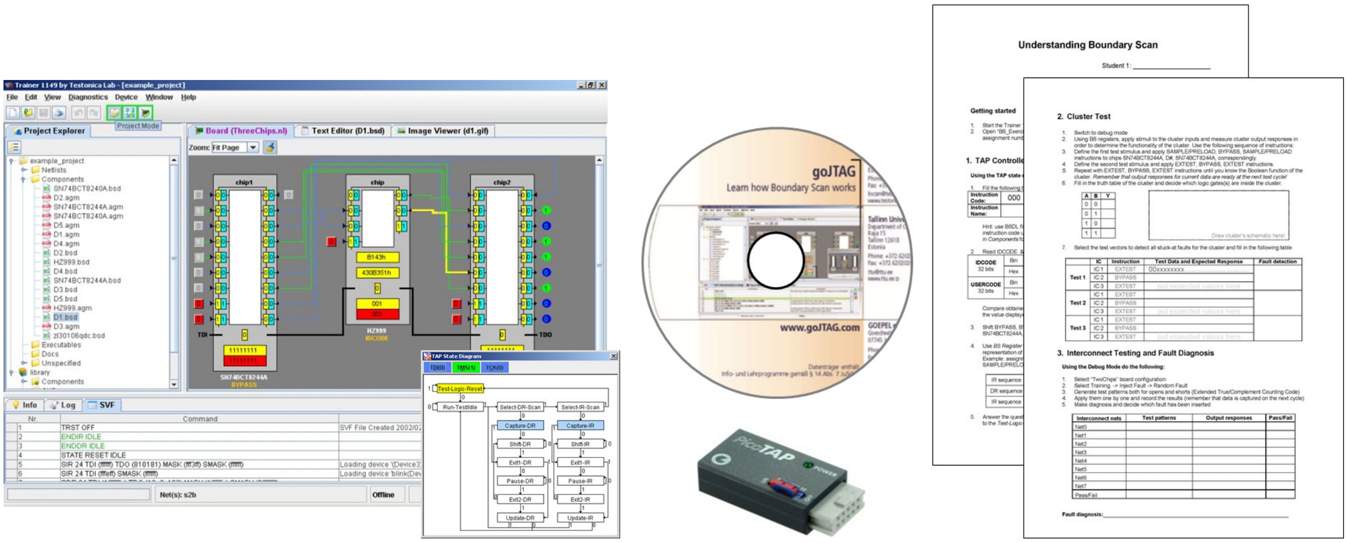
II. Overview of the Training Bundle

Trainer 1149 is the central part of the recent goJTAG initiative [6] that aims at creating an open-source platform for learning and working with Boundary Scan technology. Software source code, technical documentation, reference design and schematic of the picoTAP BS controller, lecture slides, handouts for exercises - all these materials are currently available as a part of goJTAG package. In the following we assume that the reader is familiar with the PCB test conceptions and the related IEEE 1149.1 standard.

The work was partially funded by EU Regional Development Fund via ELIKO TAK project and Centre CEBE as well as Estonian Science Foundation grants 7894 and 9423.

K. Shibin, S. Devadze, and V. Rosin are with Testonica Lab Raja 15, 12618 Tallinn, Estonia ({konstantin; sergey; slava}@testonica.com).

A. Jutman and R. Ubar are with Department of Computer Engineering Tallinn University of Technology Tallinn, Estonia ({artur; raiub}@ati.ttu.ee).



(a) (b) (c)

Fig. 1. Main components of goJTAG package: software, hardware, training materials: a) Trainer 1149 software, b) PicoTAP USB/JTAG controller, c) set of exercises.

A. Trainer 1149 Software

The main idea of Trainer 1149 is to provide an open training and simulation environment where the user can create or import own examples/projects or even experiment with industrial integrated circuits by loading Boundary Scan descriptions (BSDL files), and saving own board configurations using simple netlist format. Besides that, the package is supplied with a built-in collection of example virtual boards and library of components.

Another important feature of the software is its capability to visualize most of the aspects of Boundary Scan-based testing. The software visualizes the board under test and BS infrastructure inside each BS-enabled integrated circuit (see Fig. 1a): instruction register, ID register, boundary register, bypass register, etc. The special visualization panel is also illustrating the data flow through the scan chains and board-level interconnects. Additional interactive panel is used to illustrate the work of TAP controller. Using the same panel it would be possible to control the operation of BS structures on the board in a fine-grain way - with a half-TCK accuracy.

Trainer 1149 gives the user an opportunity to better understand the ideas behind IEEE 1149.1 standard by performing interactive experiments with virtual boards and integrated circuits. The application has the following main features implemented:

- Illustration of operation of BS registers
- Simulation of TAP Controller operation
- Injection and further diagnosis of interconnect faults (shorts, opens, etc.)
- Design or modification of BS structures inside the target chip using the BSDL
- Design or modification of boards containing several BS-enabled chips

B. USB to JTAG Controller/Cable

Trainer 1149 is capable of working with several commonly used JTAG cables, like e.g. Xilinx Parallel Cable III. Such cables are often available in laboratories dealing with electronic equipment. In addition to that, we have developed our own Universal Serial Bus (USB) to JTAG cable based on FT2232H chip from FTDI [7]. Such a cable based on a reference design gives an independence and opportunity for teachers, students and researchers to adapt the solution to their needs both cost and functionality-wise. The controller has the following features:

- TCK frequency up to 30MHz
- USB host PC interface
- Support for different target voltage levels
- Auxiliary IO signals
- configurable logic level voltages: JTAG standard does not specify exact voltages and various devices under test may have different levels
- possibility to isolate JTAG and IO signals (set outputs to high-impedance state)

This controller is seamlessly integrated with Trainer 1149 by means of a software library that interfaces generic hardware model within Trainer 1149 to FT2232H chip. The library uses D2XX library from FTDI to manipulate hardware and hides all this complexity from Trainer 1149. The designed controller was successfully tested with Trainer 1149 and real target boards. Having both hardware and software components of the bundle, it is possible to seamlessly manipulate BS function of real chips on boards from Trainer 1149.

FT2232H is a good candidate for implementing JTAG controller since it has special feature called Multi-Protocol Synchronous Serial Engine (MPSSE) which allows to implement many synchronous serial protocols like I^2C, SPI, JTAG. There are two MPSSE modules in one FT2232H chip, both capable of interfacing synchronous protocols at frequencies up to 30MHz. Thanks to internal First-In First-Out (FIFO) buffer (4 KB for incoming and outgoing data), continuous data flow can be maintained at full speed provided that software can handle it quickly enough. Besides JTAG signals, FT2232H chip can handle General Purpose Input/Output (GPIO) signals that are useful in implementing additional controller features (e.g. output disable, various control signals). Another important feature of this chip is that host interface is USB, which makes it possible to build whole JTAG controller on a single chip.

FT2232H can be controlled in host computer by means of using proprietary but royalty-free drivers. Two working modes are supported: Virtual COM Port and direct communication. In first one, the chip is seen to OS and programs as plain serial COM port while in direct communication mode additional features like MPSSE are accessible. In this mode, communication with chip is performed through additional software library provided by FTDI. These libraries provide API for other programs to control the chip.

Several important FT2232H chip properties and features are listed below:

- Two MPSSE modules with independent ports and clock generators (clock frequency up to 30MHz)
- USB 2.0 High Speed (480Mbps) and Full Speed (12Mbps) compatible
- Supported by royalty-free FTDI D2XX drivers which provide low-level access to chip including MPSSE modules
- Two sets of general purpose IO signals accessible when using MPSSE mode
- Can be powered from USB

To comply with electrical specifications and make controller capable of interfacing boards with various logic levels, voltage level shifters are introduced to the design. The function of level shifter is to electrically interface two devices when their logic levels make it impossible to connect these devices directly.

As an off the shelf solution, GÖPEL Electronic offers a JTAG controller called PicoTAP (see Fig. 1b), which is based on FT2232H chip and which is compatible with Trainer 1149.

We have compared the performance of USB-to-JTAG cable with the professional Boundary Scan controller (see Table I). Although, there is a noticeable gap in performance on DR scans with small length, on long DR scans, our controller is capable to achieve even higher data rate than the professional hardware. However, the performance is not an issue for studying Boundary Scan or carrying out laboratory works.

Also it should be noted, that in contrast to professional BS test equipment that is typically closed and can only be used with proprietary software, USB-to-JTAG cable includes open API and can be integrated into any kind of Boundary Scan software.

C. Set of Laboratory Exercises

Trainer 1149 is delivered with a set of 12 virtual boards. Ten of these boards are specifically developed for a training workflow for students. Together with slides and fill-out forms (see Fig. 1c) they represent an easy to follow tutorial explaining main concepts of BS. At the same time, the student learns Trainer 1149 software to become ready to use it later for advanced purposes.

TABLE I
COMPARISON OF IN-HOUSE USB TO JTAG CABLE WITH PROFESSIONAL EQUIPMENT

Features		USB-to-JTAG cable	Professional BS controller	Difference (times)
Software / API		free / open	proprietary	-
Cost		low	very high	200
Performance				
1000 DR-Scans of 64bit length each	1MHz	68Kbps	484Kbps	7
	30MHz	75Kbps	1041Kbps	13.9
1K DR-Scans of 1Kbit length each	1MHz	453Kbps	735Kbps	1.6
	30MHz	838Kbps	2308Kbps	2.8
Single long DR-Scan (7.5MBit length)	1MHz	811Kbps	727Kbps	0.9
	30MHz	3655Kbps	2615Kbps	0.7

The exercises are classified into 3 separate topics where every topic contains a list of tasks to perform:

1) *Learning basic concepts and principles of operation*
   - *Studying TAP Controller and TAP State Diagram*
   - *Manipulating the TMS, TDI, TDO and TCK signals*
   - *Learning BS instructions and Shift IR mode*
   - *IDCODE, BYPASS, SAMPLE, EXTEST instructions*
   - *Mapping BSDL constructs with actual structures*
   - *Shifting test data in and out of the scan chain*
   - *Controlling external LEDs from BS register*
2) *Performing Cluster Test*
   - *Driving test stimuli from BS to external devices*
   - *Reading responses from external devices*
   - *Interpreting received diagnostic data*
   - *Building truth-tables of unknown logic*
   - *Testing stuck-at faults in cluster logic*
3) *Interconnect Test and Fault Diagnosis*
   - *Interconnect fault detection principles*
   - *Test generation algorithms for target faults*
   - *Application of test patterns and diagnostic data analysis*
   - *Localization of structural faults based on test responses and fault models*
   - *Distinguishing between different classes of faults (e.g. opens, shorts, stuck-at faults)*

D. Comparison to Similar Educational Packages

There are two similar educational systems on Boundary Scan standard: Scan Educator developed by Texas Instruments in the beginning of 90-s [8] and a more recent software by GÖPEL Electronic called BScan Coach [9]. The functionality of both systems is rather limited compared to ours (see Table II).

The most important element, which is missing in both systems, is the possibility of editing existing examples and

TABLE II
COMPARISON OF SIMILAR AVAILABLE SYSTEMS

	Trainer 1149	ScanEducator	BScan Coach
Platform	Multiplatform	DOS	Windows
Usage	web/local	local	local
Chip editing	yes	no	no
Board editing	yes	no	no
Fault diagnosis	yes	no	no
Various fault models	yes	no	no
Automatic TG	yes	no	no
Test programming	yes	no	no
Import/export	yes (BSDL, SVF)	no	no
Built-in help/tutorial	basic help	splendid	nice/limited
Built-in examples	many/extendable	few/fixed	one/fixed
Hardware support	yes	no	yes, comes with a demo board
Distribution	Open source	Freeware	Freeware + paid HW

creating own ones. Import/export capability allows for industrial BS-enabled ICs to be imported into the application and then simulated. At the same time, created test programs can be exported using Serial Vector Format (SVF) that is widely accepted by the industry.

Another important difference is that the fault insertion and fault modeling features enable realistic diagnostic tasks to be performed by students for better studying of interconnect fault models as well as fault detection and localization techniques.

Moreover, Trainer 1149 contains a built-in library of different ICs, various examples of board schematics, executable files etc. This provides an opportunity to select the most illustrative example for a particular topic. One can also design his own projects right inside the system or add new collections of files to the library. The user can always import files to the newly created project or transfer them from another project.

III. Working with Trainer 1149

The main window of the application (see Fig. 1a) should give a general impression about functionality of the training system. The window is divided into three parts: control panel on the left side, set of interactive viewing panels for different file types and the bottom panel. We also define three working modes (each of them is associated with a group of dedicated windows and panels): *Project Mode* (Fig. 1a), *Debug Mode* and *Board Edit Mode*.

A. Main Components of GUI

The board under test is shown in the central part of the main window called the *Board Viewer*. It supports two representations of the loaded board: *System Overview* (Fig. 1a) and *TAP Chain* (Fig. 2). In *System Overview* all supported components are shown in the graphical panel as well as all connections between these components. Some of supported components like BS chips have also parts of internal structure that are illustrated. Details of *System Overview* outlook will be described later.

The *TAP Chain* representation is limited to represent BS scan chain elements only. The goal of this outlook is to show schematically a chain of BS components connected from TDI to TDO. Representation of BS-supported components is also very simplistic (Fig. 2): just a small gray rectangle with no internal structures neither interconnects (except TDI-TDO line).

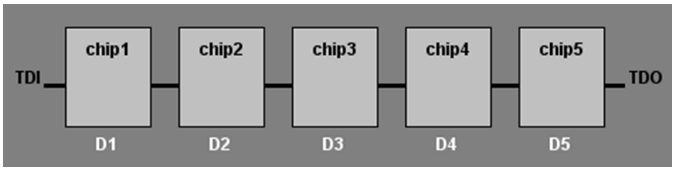


Fig. 2. TAP chain view.

In the *System Overview* BS chips are shown in a much more detailed manner (Fig. 1a). The white area inside the chip represents its core logic, which is not illustrated in detail. The wrapper around the core logic is the *Boundary Scan Register* (BSR). Each scan cell of this register consists of two flip-flops. The one marked by yellow color (also called the *shift part*) is used for capturing the state of the corresponded test point. Another one – marked by light-green (also called the *update part*) is needed for keeping the driving value of the test point during the test mode. The test data and captured responses are shifted in and out in series via TDI/TDO pins and the shift part of the BSR.

Highlighting signals by different colors is intended for making it easier to follow the simulation. Any changes in the state of a board can be quickly noticed visually.

When cells of the BS register are shown inside the chips it is also possible to click on them. This will make a *Component Details* window visible. The *Component Details* window contain an image of possible logic-level schematic of selected cell. Ten BS types are defined by the standard are supported in Trainer 1149: BC_1 through BC_10.

The register shown at the very bottom of the chip is the *Instruction Register* (IR). It shows the current active BS instruction (in binary form) in the chip (see Fig. 3). Similarly to BSR, the IR has two parts. The next instruction is shifted using the upper part of IR shown in yellow. After the instruction has

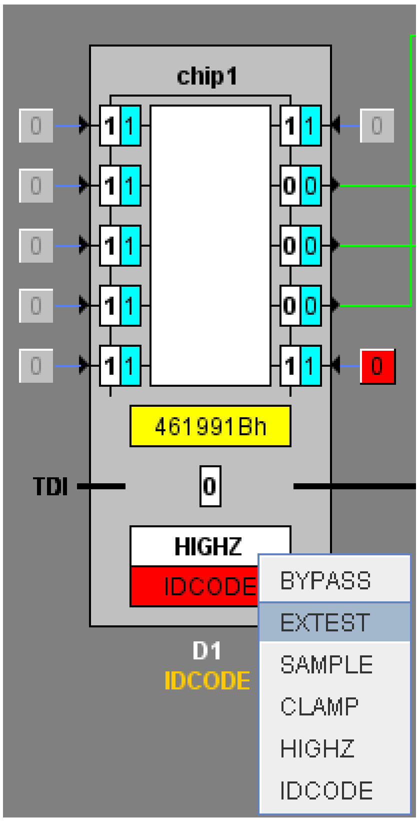
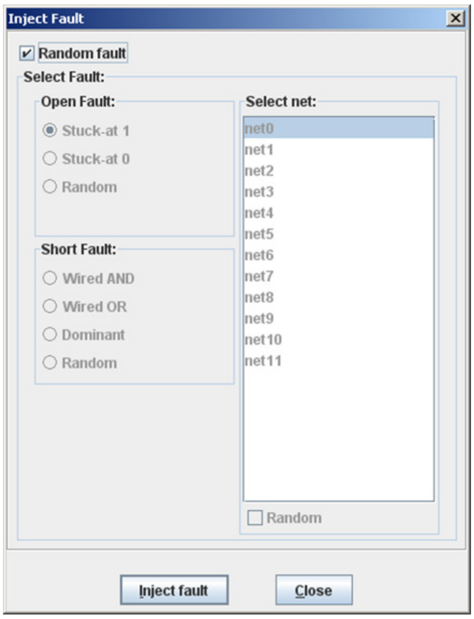


Fig. 3. Test pattern insertion mode.


Inject Fault

☑ Random fault

Select Fault:

Open Fault:

● Stuck-at 1
○ Stuck-at 0
○ Random

Short Fault:

○ Wired AND
○ Wired OR
○ Dominant
○ Random

Select net:

net0
net1
net2
net3
net4
net5
net6
net7
net8
net9
net10
net11

□ Random

Fig. 4. Inject fault dialog window.

been completely shifted in, the lower part is updated with the values from the upper part. The one-bit register just above the Instruction Register is the *Bypass Register*. It is activated by BYPASS instruction and usually used for faster test data shifting.

The Instruction Register and the Bypass Register are mandatory TDRs. Although BSDL description can also contain some optional design-specific TDRs, they will not be shown graphically. The only optional registers that will be illustrated are the device *ID Register* and device *User Code* register.

Logic clusters are represented as grey rectangles with the name of cluster and signal names written upon them. Logic clusters can be also simulated if the internal logic is described in a compatible format. For this purpose we have selected the SSBDD (Structurally Synthesized Binary Decision Diagram) logic-level description format [10] because of its simplicity and efficiency. This format can be easily converted from a widely-used EDIF description by Turbo Tester software [11]. No visualization of such simulation is provided as clusters do not contain any BS infrastructure.

Board input pins are represented as clickable buttons. Ordinary data input pins are grey, while the control pins are red. By clicking the pins 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.

Board output pins are represented as indicators (colored circle with a value, mimicking a LED), showing the current state of the output line. Logic 1 and logic 0 are indicated by green and blue colors correspondingly. The wires driven into the high impedance state are white. In last case, the output indicator will hold the value “Z”. If a wire is red or the indicator shows “X”, then the logic value of the corresponding signal is unknown. Highlighting signals by different colors is intended for making it easier to follow the simulation. Any changes in the state of a board can be quickly noticed visually.

The test bus wire connecting all the TDIs and TDOs of all the chips together is displayed by a bold black line. This bus is used for test data exchange with an external tester.

B. Fault Injection Mechanism

Using *Training* menu it is possible to inject various types of interconnect faults into currently selected board (see [12] for details). The supported fault models are stuck-at fault model, wired-and, wired-or and dominant short fault models.

The type of fault to be injected is selected using *Inject Fault* dialog window (Fig. 4). It is also possible to insert a random fault. Also the faulty net (or two nets for wired/dominant fault models) can be selected from the provided list or randomly.

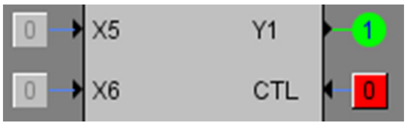


Fig. 5. Signals, buttons, LEDs.

After the fault is inserted, all signals in Board Viewer are losing value-specific color (become black). By applying the sequence of tests, the user should try to find the faulty signal and identify the type of fault. After the fault is identified, the diagnosis can be verified using *Training* → *Check Fault* menu item. To remove the injected fault from the board *Training* → *Remove Injected Faults* menu item should be used.

If the fault is injected into the board the corresponded mark is shown on the status bar.

C. Fault Diagnosis and Test Pattern Insertion

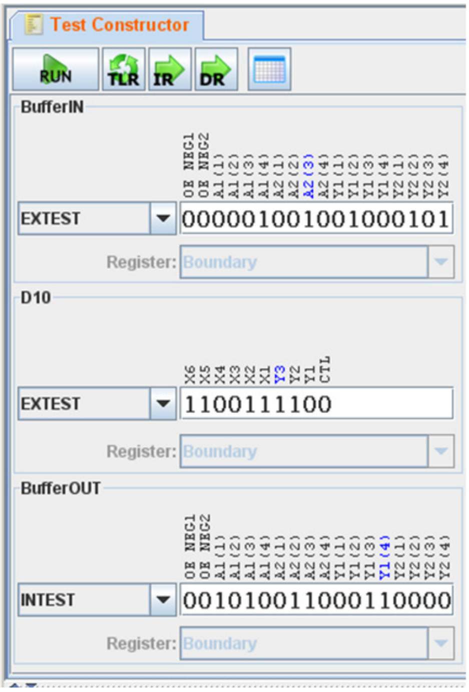
The fault diagnosis can be performed by entering a special *Debug Mode*. It combines different possibilities of test pattern insertion, applying and analyzing obtained results.

There are two ways how the test data can be manually inserted:

- using *Test Constructor* panel
- directly in the *Board View* panel

Both possibilities are synchronized between each other. Therefore, the values of test data vector in *Test Constructor* panel and *Board View* always match each other. Also there are controlling buttons which start instruction or data shifting processes. They are located in the toolbar of *Test Constructor* panel (that can be also considered as a control panel of the *Debug Mode*).

Test Constructor panel is shown in Fig. 6. It contains lists of available BS instructions for each chip. These instructions are defined in the BSDL description. On the right side from each list a special input fields are located. These fields are used for entering test vectors. The vector inside this field (it is a part of whole test pattern) will be applied for specific chip. Only 0 or 1 are allowed as a valid input and the length of input data should be equal to the length of currently selected register in a chip. Above these input fields there are list of names of signals connected to BS cells (in case of boundary register) or bit ordering numbers (in case of any other register).



BufferIN

OE NEG1, OE NEG2, A1(1), A1(2), A1(3), A1(4), A2(1), A2(2), A2(3), A2(4), Y1(1), Y1(2), Y1(3), Y1(4), Y2(1), Y2(2), Y2(3), Y2(4)

EXTEST ▾ 000001001001000101

Register: Boundary ▾

D10

X6, X5, X4, X3, X2, X1, Y3, Y2, Y1, CTL

EXTEST ▾ 1100111100

Register: Boundary ▾

BufferOUT

OE NEG1, OE NEG2, A1(1), A1(2), A1(3), A1(4), A2(1), A2(2), A2(3), A2(4), Y1(1), Y1(2), Y1(3), Y1(4), Y2(1), Y2(2), Y2(3), Y2(4)

INTEST ▾ 001010011000110000

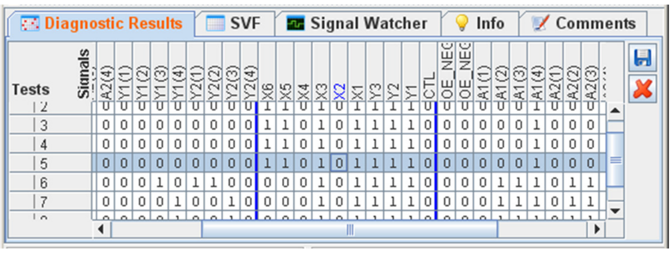
Register: Boundary ▾

Fig. 6. Test constructor panel.

Another possibility is to edit test data directly on the virtual board. In the Debug Mode special white fields appear inside boundary scan cells, instruction registers and bypass registers (Fig. 3). These fields display the value or bit-vector that will be shifted into the corresponding cell/register during the next test data shift. The user can easily modify such a test vector by toggling certain cells or selecting values from a drop-down list. The same principle holds for IR and Bypass register.

When the test data is prepared, the user has to apply the test to the board in the following way. Pressing "*Scan IR*" button, makes selected instructions to be applied to the corresponding ICs. Then, the user should press "*Scan DR*" button to force all the input data to be shifted in. Clicking "*Run*" button is equivalent to subsequent performing of both "*Scan IR*" and "*Scan DR*" operations.

During the data shift-in operation, the current state of selected Data Registers is shifted out. The resulting TDO bit sequence (vector) will be shown in the *Diagnostic Results* panel (Fig. 7). *Diagnostic Results* panel has a table-based structure, where one row represents one test vector. For every test pattern there is an expected output response (computed by simulator). If the response is different from the expected one, the fault has been detected. Input and output vectors are automatically compared and differences are shown with red color. The result table can be saved into a file or printed out at any time.



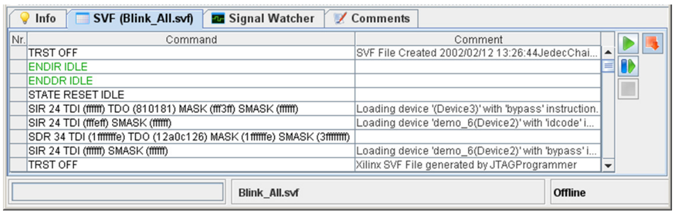
Tests / Signals	A2(4)	Y1(1)	Y1(2)	Y1(3)	Y1(4)	Y2(1)	Y2(2)	Y2(3)	Y2(4)	X6	X5	X4	X3	X2	X1	Y3	Y2	Y1	CTL	OE_NEG	OE_NEG	A1(1)	A1(2)	A1(3)	A1(4)	A2(1)	A2(2)	A2(3)
2	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
6	0	0	0	1	0	1	1	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1	1	0	1	1
7	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1	1	0	1	1

Fig. 7. Diagnostic results panel.

D. Connecting Hardware and Working with SVF Files

Trainer 1149 has a full-featured support of Serial Vector Format (SVF) that was designed for exchanging descriptions of high-level Boundary Scan bus operations. It is widely used for storing test and configuration programs for boundary scan devices.

The system provides two possibilities for editing SVF files. The first option is to enter commands line by line using native SVF syntax. Entered information will be automatically analyzed by SVF parser in order to detect possible syntax and semantic errors. Fig. 8 shows the SVF panel with a fragment of test program.



Nr.	Command	Comment
	TRST OFF	SVF File Created 2002/02/12 13:26:44JedecChai...
	ENDIR IDLE	
	ENDDR IDLE	
	STATE RESET IDLE	
	SIR 24 TDI (ffffff) TDO (810181) MASK (fff3ff) SMASK (ffffff)	Loading device '(Device3)' with 'bypass' instruction.
	SIR 24 TDI (fffeff) SMASK (ffffff)	Loading device 'demo_6(Device2)' with 'idcode' i...
	SDR 34 TDI (1fffffffe) TDO (12a0c126) MASK (1ffffffe) SMASK (3ffffffff)	
	SIR 24 TDI (ffffff) SMASK (ffffff)	Loading device 'demo_6(Device2)' with 'bypass' i...
	TRST OFF	Xilinx SVF File generated by JTAGProgrammer

Blink_All.svf

Offline

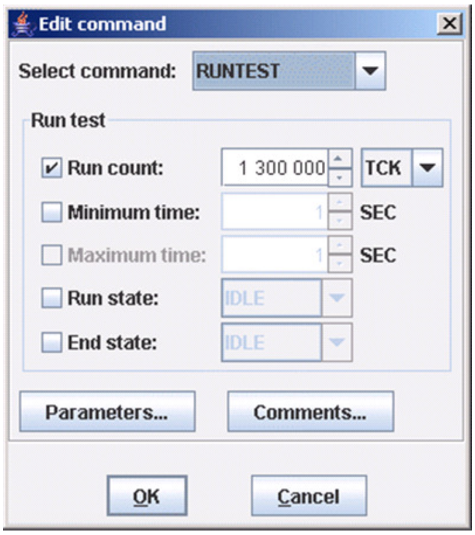
Fig. 8. SVF editor panel.

Another way is to use a dedicated GUI for composing test program. In the last case, it is possible to specify commands that define actions (*STATE*, *SDR* and others) by entering info in the special dialog window (Fig. 9). If user specifies additional parameters for execution (such as *FREQUENCY*, *ENDIR*, *TDR* and others) the corresponded SVF commands will be inserted automatically.

Execution of SVF files can be performed in non-interactive and interactive modes. In the interactive mode, user can execute commands step-by-step (or perform execution until specified line) and then observe the reaction of simulated board on each step. In the non-interactive mode user just runs test program and waits until it is finished to get the results of execution. After the execution, the software analyzes actually received output and compares it with the expected one. Commands that produce invalid output will be marked by red color as failed ones. After clicking on such commands user will get the detailed information about the failure. There is also a possibility to interrupt program execution after the first failure.

Note that if the board is connected to the application during SVF execution, the software will pass the signals and read the response directly from the plugged device instead of using software simulation. This gives user the opportunity to compare results of software simulation and real behavior of hardware.

The latest version of software is equipped with a high-performance USB to JTAG cable that allows to work with real demonstration hardware to encourage students to perform their experiments in a realistic environment.



Edit command

Select command: RUNTEST

Run test

☑ Run count: 1 300 000 TCK

□ Minimum time: 1 SEC

□ Maximum time: 1 SEC

□ Run state: IDLE

□ End state: IDLE

Fig. 9. SVF command edit window.

IV. Conclusions

In this paper, we have described a multi-purpose system, which provides a simulation, demonstration, and CAD environment for learning, research, and development related to IEEE 1149.1 Boundary Scan standard.

A BS device manipulation is quite a tricky exercise. Therefore, only a system, which allows instant simulation and illustration of all the users steps can help learning and easy finding all possible mistakes and misunderstandings, which otherwise would likely be missed out.

Trainer 1149 is equipped with the following important working modes like fault modeling, functional simulation, manual test vector generation, and import/export interface. The most important of all these improvements has to do with test generation and application.

The system has also been supplied with a short introductory description, dedicated exercises for students and a USB cable.

Acknowledgment

Authors would like to thank colleagues from TU Ilmenau and GÖPEL Electronic GmbH for valuable collaboration and contribution to the creation of the training package.

References

[1] *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE Std. 1149.1-2001, 2001.

[2] (2010) The International Technology Roadmap for Semiconductors, 2010 Update: Test and Test Equipment. [Online]. Available: http://www.itrs.net/

[3] D. Crosier, L. Purser, and H. Smidt, "Trends V: Universities Shaping the European Higher Education Area," European University Association, p. 97, 2007.

[4] (2011) Trainer 1149. Testonica LAB. [Online]. Available: http://www.testonica.com/1149/

[5] H.-D. Wuttke and K. Henke, "Teaching digital design with tool-oriented learning modules "living pictures"," in *Proc. of 32nd ASEE/IEEE Frontiers in Education Conference*, vol. 3, Boston, USA, nov. 2002, pp. S4G–25 – S4G–30.

[6] (2011) goJTAG initiative. [Online]. Available: http://www.gojtag.com/

[7] "FT2232H dual high speed USB to multipurpose UART/FIFO IC," data sheet FT_000061, Future Technology Devices International Limited, 2011.

[8] (2011) JTAG Scan Educator. Texas Instruments Incorporated. [Online]. Available: http://www.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=satb002a

[9] (2011) JTAG/Boundary Scan Coach - interactive learning software for IEEE1149.X. GOPEL Electronic. [Online]. Available: http://www.goepel.com/index.php?L=4&id=1418

[10] A. Jutman, A. Peder, J. Raik, M. Tombak, and R. Ubar, "Structurally synthesized binary decision diagrams," in *Proc. of 6th International Workshop on Boolean Problems (IWSBP'04)*, Freiberg, Germany, 23–24 Sep. 2004, pp. 271–278.

[11] (2011) Turbo tester. [Online]. Available: http://www.pld.ttu.ee/tt/

[12] A. Jutman, R. Ubar, and V. Rosin, "A Software System for IEEE 1149.1 Boundary Scan Design, Simulation, and Demonstration," in *IEEE European Board Test Workshop*, Tallinn, Estonia, 25–26 May 2005.