Understanding Boundary Scan Test with Trainer 1149

A. Jutman[1], S. Devadze[2], K. Shibin[1], V. Rosin[2], R. Ubar[1]

[1]Tallinn University of Technology, Estonia
[2]Testonica Lab, Estonia

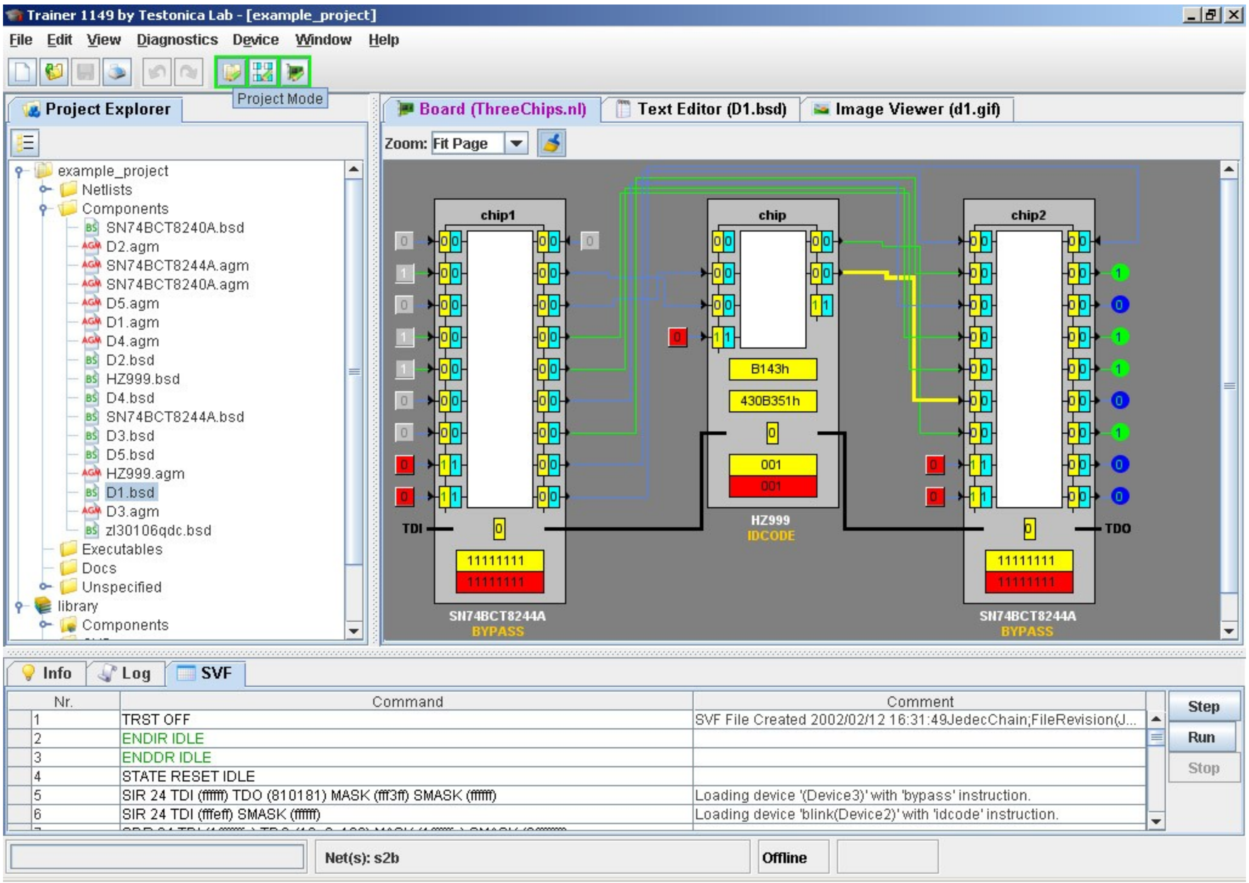
Abstract

In this paper, we describe a multi-functional software system, which provides a simulation, demonstration, and training environment for learning, research, and development related to IEEE 1149.1 Boundary Scan (BS) standard. The system supports important Boundary Scan data formats (BSDL, SVF) through which it can interact with other BS development tools. At the same time, the system provides a graphical design and simulation environment of BS-enabled chips and non-BS clusters. It is also possible to simulate the behavior of various interconnect faults and inspect them using interactive tools. Using a convenient low-cost USB-JTAG cable on can test real defects in real hardware. The system is implemented using multi-platform Java environment and distributed as a freeware. Such combination of features is unique for public domain BS software.

1. Introduction

The modern society nowadays experiences rapid changes driven by technological advances – in the first place, in microelectronics and information technologies. This trend affects inevitably general education styles and learning habits. At the same time, it creates a demand for state-of-the-art technology-related topics and courses in the curricula. The current paper contributes to both mentioned aspects by introducing a new training/learning tool in the field of microelectronics reliability.

Accordingly 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 [1]. The



Nr.	Command	Comment
1	TRST OFF	SVF File Created 2002/02/12 16:31:49JedecChain;FileRevision(J...
2	ENDIR IDLE	
3	ENDDR IDLE	
4	STATE RESET IDLE	
5	SIR 24 TDI (ffffff) TDO (810181) MASK (fff3ff) SMASK (ffffff)	Loading device '(Device3)' with 'bypass' instruction.
6	SIR 24 TDI (fffeff) SMASK (ffffff)	Loading device 'blink(Device2)' with 'idcode' instruction.

Figure 1. Main Window of the System (Project Mode)

Table 1. 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

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, the new tool 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.

The training tool was designed accordingly to the concept of “Living Pictures” [2]. 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.

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 [3]. The state of the art PCB testing is a mixture of Boundary Scan (BS), optical 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 in in-circuit programming and product maintenance.

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

2. System Overview

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

The board under test is shown in the central part of main window. In our example, it consists of three BS-enabled chips connected into a single chain. In addition, board is shown together with all active signals passing through interconnects and BS registers. However, in the *Board Edit* working mode user can always reconfigure BS chain by adding new components from the built-in library or removing the existed ones.

Bottom part of a tool contains several panels, that represent different kind of application output: information panel (contains detailed information about project files), diagnostic panel (used in fault diagnosis process) and log panel (logs informational and error messages).

There are also *Component Outlook* pop-up window (illustrates detailed structure of different types of BS register cell) and *Outlook* window (that helps in navigation through complex boards with a large number of elements). In addition, TAP controller state diagram can be brought on screen at any moment.

There are two similar educational systems on Boundary Scan standard: Scan Educator developed by Texas Instruments in the beginning of 90-s [3] and a more recent software by GÖEPEL Electronic called BScan Coach [6]. The functionality of both systems is rather limited compared to ours (see Table 1).

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

Another important difference is that the fault insertion and fault modeling features, enables real diagnostic tasks to be performed by students for, better studying of interconnect fault models and fault detection and diagnosis methods.

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.

3. Fault Diagnosis and Test Pattern Insertion

There are basically two simulation modes supported by the application: fault-free simulation and simulation for fault detection. The latter option implies that there is a certain faulty connection or group of them. Then the goal of this simulation is to find these faulty connections and identify the type of fault. This is done by selection of proper BS instructions and input test data.

Using *Diagnostics* menu it is possible to insert various types of interconnect faults into currently selected board (see [4] for details). The fault diagnosis can be performed by using special *Test Pattern Insertion Mode* . It combines different possibilities of test pattern insertion, applying and analyzing of obtained results.

Let us start with creating of test data. The first way to do this is to use *BS Command* panel. It contains lists of available BS instructions for each chip. These instructions are defined in the BSDL description and could vary for from chip to chip. The user just selects required instructions for the chips and defines the input data in corresponded text fields.

Another possibility is to insert test data directly on the virtual board. After changing to *Test Pattern Insertion Mode* graphic representation of the board inside the *Board Viewer* panel also becomes oriented to test data insertion: special white areas appear inside boundary scan cells, instruction registers and bypass registers. The white area inside the specific cell shows value, that will be shifted into this cell during next test data loading. So, cell by cell these values form one test vector. User can easily modify this test vector by inverting the value of the correspondent cell just by clicking on it. The name of BS instruction, which will be loaded into chip, is shown inside the white rectangle of instruction register. To select or modify the instruction user just chooses it from the list, that appears after clicking on the white area of instruction register. In the same way clicking the bypass register inverts the value that will be loaded into it during serial shift-in operation.

Both described possibilities are interconnected with each other, i.e. test data vector has the same value on the *BS Command* panel and inside virtual BS cells.

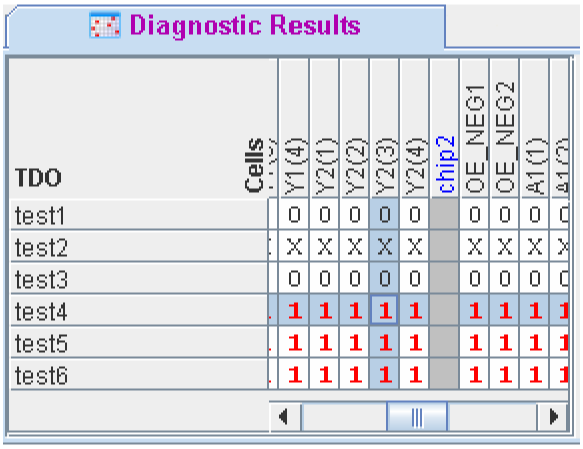
After test data definition procedure is finished, user needs to supply test to the board. By pressing "*Scan IR*" button of *BS Command* panel selected instructions will be applied to the corresponded chips. Next, user can press "*Scan DR*" button to force all the input data be shifted in. Clicking "*Run*" button is equivalent to sequential performing of "*Scan IR*" and "*Scan DR*" operations.

After test was successfully applied to the board, the output results (data that were shifted out) will be shown in the *Diagnostic Results* view (Figure 2). Every row inside the table represents one test vector. For an 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. In the current version of the application there is no need to analyze the output data manually. Input and output vectors are automatically compared and differences are shown with red color inside the table of *Diagnostic Results* panel. The result table can be saved into file or printed out at any moment of diagnostic procedure.

4. Connecting hardware and working with SVF files

We have also added a full-featured support of Serial Vector Format (SVF) to our application. SVF format that was designed for exchanging descriptions of high-level Boundary Scan bus operations and is widely used for creating test programs for boundary scan devices.

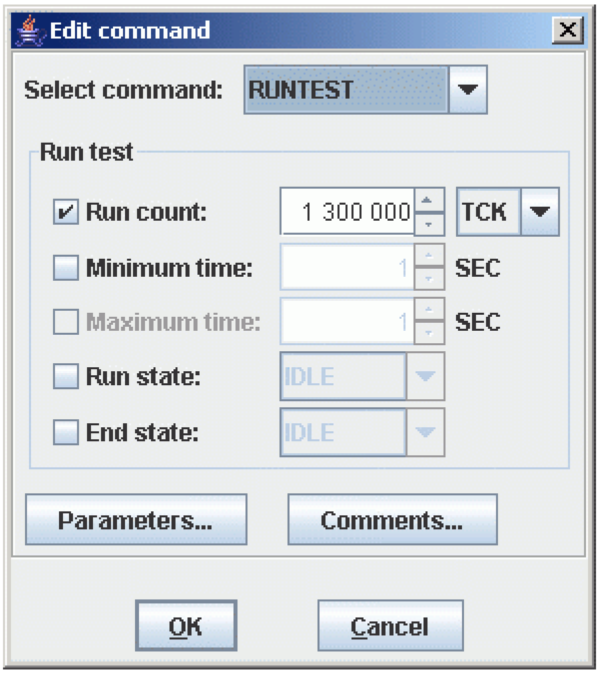
Execution of SVF files can be performed in non-interactive and interactive modes. In the interactive mode, user can execute commands step-by-step (or perform execution until specified line) and then observe the reaction of simulated board on each step. In the non-interactive mode user just runs test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



TDO	Cells	Y1(4)	Y2(1)	Y2(2)	Y2(3)	Y2(4)	chip2	OE_NEG1	OE_NEG2	A1(1)	A1(2)
test1		0	0	0	0	0		0	0	0	0
test2		X	X	X	X	X		X	X	X	X
test3		0	0	0	0	0		0	0	0	0
test4		1	1	1	1	1		1	1	1	1
test5		1	1	1	1	1		1	1	1	1
test6		1	1	1	1	1		1	1	1	1

Figure 2. Diagnostic Table

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.



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

Figure 3. SVF Command Edit Window

This can be done using the dedicated USB to JTAG cable (and controller) that we have developed specifically for Trainer 1149. The cable is based on FT2232H chip from FTDI.

At low (signal) level, JTAG controller needs TAP interface to connect to JTAG targets. At this level, hardware solution is preferable since it provides higher data throughput and lower latency. At the same time, controller should have widespread host interface such as USB. Both requirements are satisfied by usage of FT2232H chip from FTDI.

FT2232H chip includes MPSSE (Multi-Protocol Synchronous Serial Engine), configurable serial controller designed to support JTAG, SPI, I2C and other synchronous serial protocols with clock frequency of up to 30MHz. Thanks to internal FIFO buffer (4KB for incoming and outgoing data), continuous data flow can be maintained at full speed provided that software can handle it quickly enough. FT2232H contains whole USB 2.0 High-Speed interface and MPSSE on a single chip, eliminating need for additional ICs and lowering total controller cost. Besides JTAG signals, FT2232H chip can implement GPIO signals that are useful in implementing additional controller features (output disable, various control signals). FTDI provides low-level access to MPSSE unit inside the chip through software library. This enables designers to suit protocol implementation to their needs and implement other and future serial synchronous protocols.

5. Conclusions

In this extended abstract, we have described a multi-purpose system [9], 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 user's 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.

6. References

[1] D. Crosier, L. Purser, H. Smidt: Trends V: Universities Shaping the European Higher Education Area. European University Association asbl, 2007, 97 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, Boston, USA, Nov. 6 - 9, 2002, Session S4G, pp. 25-30.

[3] IEEE 1149.1-1990, "IEEE Standard Test Access Port and Boundary-Scan Architecture", 1990.

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

[5] Texas Instruments' Scan Educator URL: http://focus.ti.com/docs/toolsw/folders/print/scan_educator.html

[6] GÖPEL Electronic Boundary Scan Coach URL: http://www.goepel.com

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

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

[9] Trainer 1149.1 URL: http://www.testonica.com/1149/