Interactive Training Tools for Basic Introduction to VHDL Models

Aneliya Ivanova

Abstract: *The paper outlines the increasing popularity of Hardware Description Languages (HDL), the advantages of VHDL and the reasons it gained an important position in engineering education. Special attention is paid to some difficulties that students meet when VHDL is first introduced to them. The good practices and recommendations are considered and an approach to help the students to overcome these difficulties is discussed. It consists in development of interactive training tools that in a visual manner will help the students to comprehend the connection "VHDL statement" – "hardware structure and behaviour". The author believes this approach will address better the present students that tend to react better to visual information instead of textual one.*

Key words: *VHDL, VHDL Model, Interactive Training Tool, Design Technology Course.*

INTRODUCTION

The more popular become the programmable logic devices, the more important becomes to introduce to students the concepts and principles of HDL. After they have been introduced, within a few years, both VHDL and Verilog emerged as the dominant HDLs in the electronics industry.

Smith [1] provides a well structured comparison of VHDL and Verilog. It could be shortly summarized as follows: hardware structure can be modelled equally effectively in both VHDL and Verilog, but when modelling abstract hardware, the capability of VHDL is higher. There are more constructs and features for high-level modelling in VHDL than in Verilog, as a result there are many ways to model circuits, especially those with large hierarchical structures. VHDL may be preferred because it allows a multitude of language or user defined data types to be used. It also allows concurrent procedure calls and reusability – in VHDL procedures and functions may be placed in a package so that they are available to any design unit. VHDL models must be coded precisely with defined and matching data types because VHDL is a very strongly typed language. This may seem difficult for novices but will develop skills for precise design and modelling of hardware.

When considering the role of VHDL in engineering education, because of its power for abstract modelling, it is very suitable for introducing the top-down design methodology to students [2]. VHDL is also "an ideal tool for students to learn about computer architecture at a detailed level because of its ability to model digital system components" [3]. At the University of Ruse, VHDL has been taught since 2004. In 2008 the lecturers' team of "Design Technology" course upgraded the curricula, lectures, lab equipment and lab assignments of the course [4] in order to increase its quality and to make it more attractive for students.

Throughout the course evolution were observed some difficulties that students meet when studying the principles and concepts of hardware design with VHDL. The main problem is that the students consider VHDL as a programming language and are not able to "see" the connection between VHDL constructs and the designed hardware.

The research shows this is not an isolated case – Wang [5] gives some useful recommendations to facilitate the introduction of VHDL. In short: when introducing VHDL to students, it is very important to point out that VHDL is NOT a programming language, but it is used for describing the required digital systems. Before VHDL is introduced it is very important and essential for the students to understand the basic principles of digital circuits and digital logic design and have a good understanding of the type and behaviour of digital logic that should be produced. The author of [5] gives a solution of the problem through preparing code examples that help the students better understand the "hardware nature" of VHDL.

This paper presents an approach to solve this problem through development of interactive training tools that in a visual manner will help the students to comprehend the connection "VHDL statement" – "hardware structure and behaviour". The author believes this approach will address better the present students that tend to react better to visual information instead of textual one.

DESIGN

The lab assignments of "Design Technology" course start with design of simple sequential and combinational logic devices (registers, counters, decoders, etc.) and finishes with complex projects (e.g. processor). The course work requires design of a digital device, consisting of several simple logic devices. The difficulties appear exactly in these two parts of the course – where students using Integrated Software Environment (ISE) develop real projects – from writing the VHDL code to configuration the programmable logic device (CPLD or FPGA). For example, if a syntax error is found in a C program, the compiler usually points where the error is. In a VHDL design one signal output could be inferred from several VHDL statements concurrently. Therefore when a problem occurs, the ISE only provides a general indication of what or where the problem could be. Students, thinking in terms of pure programming, spend a lot of time trying to find the problem, because they do not have sufficient understanding about the behaviour of the hardware they model with VHDL. This is precisely the main purpose of the training tools discussed below – to create namely this understanding.

The first design issue is focused on the user analysis. The designed training tools are targeted to two general groups of students: those who are developing their lab assignment and those who are developing their course project. Taking into account the introductory discussions, the following conditions of use of the tools are outlined:

- the users have a knowledge gap in the field of digital circuits or digital logic design;
- the users are not able to understand the VHDL concepts (abstraction levels, data types, expressions and operators);
- the users are not able to connect the notion of VHDL statements to the notion of the structure and behaviour of the designed hardware.

The next issue is focused on the structure of the designed application. Considering the course content and structure, it is reasonably to separate the training tools thematically in two applications that introduce the VHDL modelling respectively of combinational and sequential logic. The first tool will illustrate various VHDL models of decoders, coders, multiplexers and comparators, the second one will include models of registers, counters and finite state machines (FSM).

The next consideration is about the user interface: here the way of interaction with the user and the way of organization of the work area is to be specified. Fig.1 displays the use case diagram generally representing "user-training tool" interaction. The diagram shows that when the student, who started the application, selects the type of the device (e.g. "decoder") and a particular variation (e.g. "3/8") hi/she is provided with 5 different modes of logic device representation, simultaneously displayed. They are: truth table; graphical notation; animation, illustrating device's behaviour (the student is able to start, pause and stop the animation); VHDL model and waveform of VHDL code simulation. When the student is not able to understand the VHDL code, he/she could read a detailed explanation of each statement of the VHDL model. When the student is interested in additional information about the selected device, detailed description of the device is provided, too. Fig.2 presents the user interface template of the tools' work area (the screen that appears when a logic device is selected from the start screen of the application).

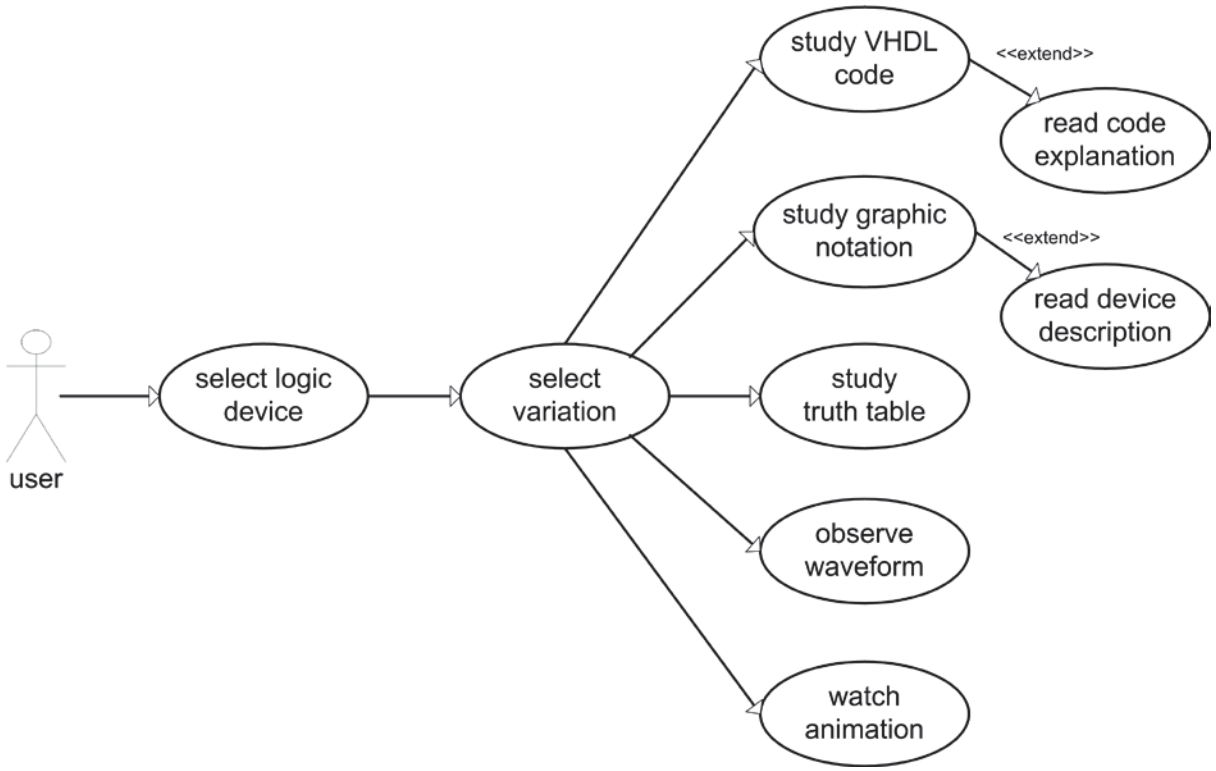

Fig.1. Interaction between the user and the training tools

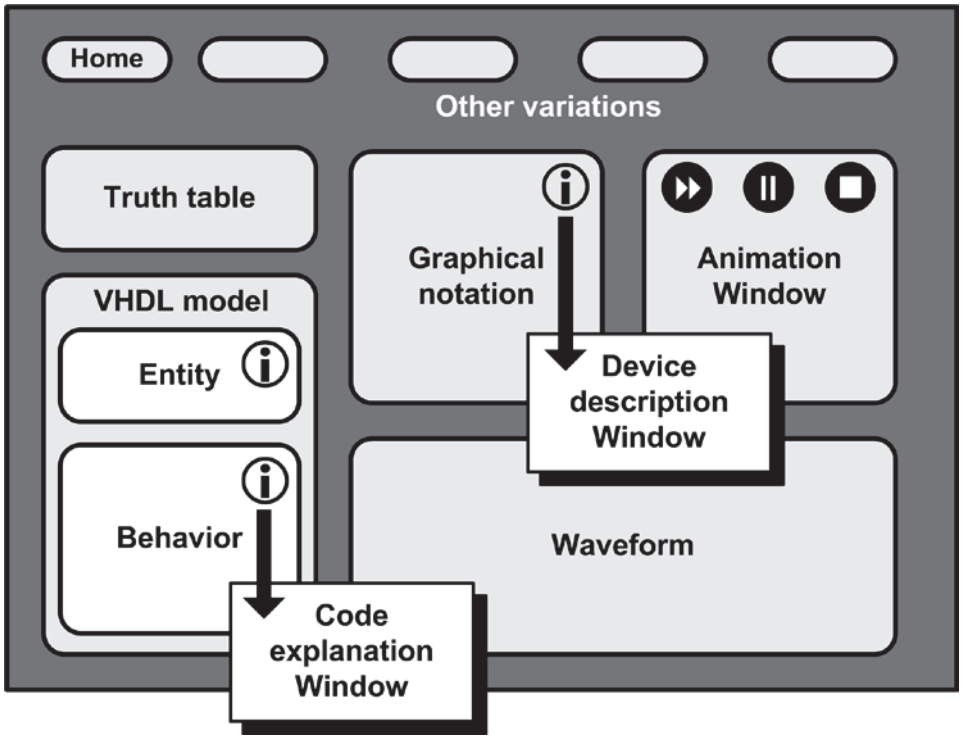

Fig.2. Work area of the training tools – user interface template

IMPLEMENTATION

Seeking after platform and location independency, the training tools are implemented using Macromedia Flash MX as for each tool two compilations are produced – executable file (for local use) and .swf format (for WEB-based use). The completed applications are

integrated in "Design Technology" WEB site and are also locally available at the lab workstations. Fig. 3 and 4 show exemplary screens from the implemented training tools.

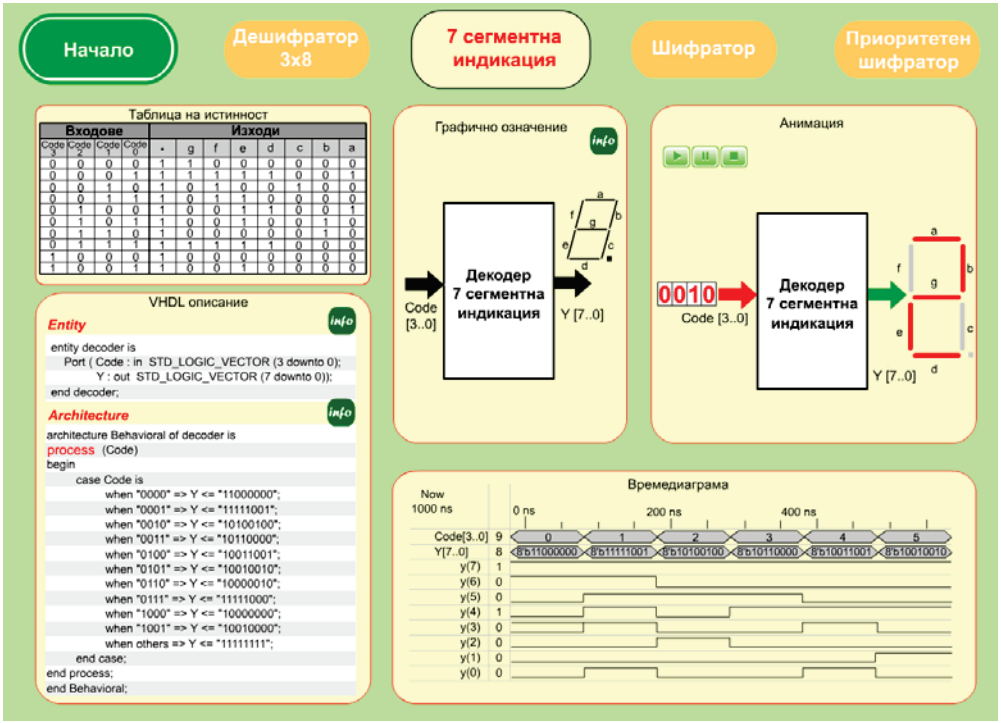

Fig.3. Introduction to VHDL models of combinational logic devices – "Decoder 7-segment indication" screen

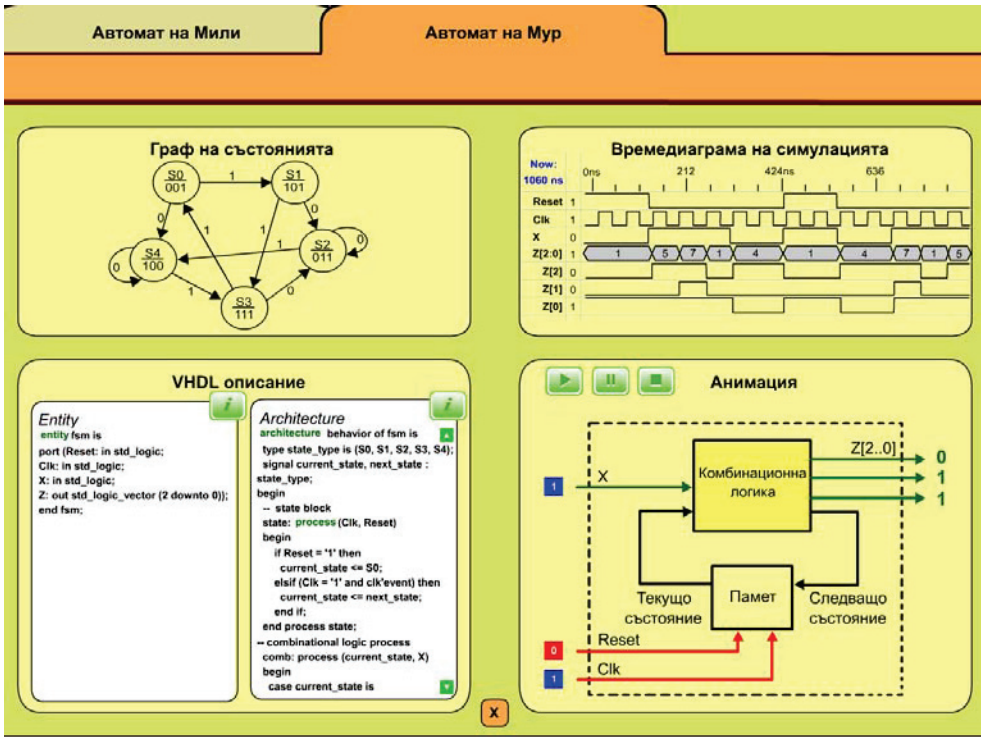

Fig.4. Introduction to VHDL models of sequential logic devices – "Moore machine" screen

CONCLUSIONS AND FUTURE WORK

The adoption of the implemented training tools within "Design Technology" course was accepted very positive by students. They shared the opinion that the tools really help them to understand better the behaviour of represented digital devices and connect the idea about this behaviour to the corresponding VHDL models. They also liked the manner of representation, namely to have "all-at-a-time" – VHDL code, statements' explanation, simulation waveform, interactive animation, truth table, etc. without necessity to look up multiple sources of information. The tools were excellent helper not only for the labs, but also for the course project development and for preparation for the control works. In the future the current results could be extended by development of training tools, introducing more complex digital devices (e.g. memories, controllers, programmable counters, etc.).

REFERENCES

[1] Smith, D.J. VHDL and Verilog compared and contrasted-plus modeled example written in VHDL, Verilog and C. Proceedings of Design Automation Conference, 1996. Available online at: http://www.angelfire.com/in/rajesh52/verilogvhdl.html

[2] Chang, J. M. Teaching Top-Down Design using VHDL and CPLD, Proceedings of IEEE Frontiers in Education Conference, Salt Lake City, Utah, November 6-9, 1996.

[3] Huang, T. C., R.W.Melton, P.R.Bingham, C.O.Alford, F.Ghannadian. The Teaching of VHDL in Computer Architecture. Proceedings of the 1997 International Conference on Microelectronics Systems Education, 1997.

[4] Ivanova, A. N. Kostadinov, G. Ivanova. Teaching VHDL for Students in Computing at the University of Rousse. ACM International Conference Proceeding Series, 2008, Vol. 374, pp. VII.1-1 – VII.1-6.

[5] Wang, G. Lessons and Experiences of Teaching VHDL. Proceedings of the 2007 American Society for Engineering Education Annual Conference & Exposition, 2007.

ACKNOWLEDGEMENT

This paper is financed by the project: Creative Development Support of Doctoral Students, Post-Doctoral and Young Researches in the Field of Computer Science, BG 051PO001-3.3.04/13, EUROPEAN SOCIAL FUND 2007– 2013 OPERATIONAL PROGRAMME "HUMAN RESOURCES DEVELOPMENT".

ABOUT THE AUTHOR

Principal Lecturer. Aneliya Ivanova, PhD, Department of Computing, University of Ruse, Phone: +359 82 888 827, E-mail: aivanova@ecs.uni-ruse.bg