VHDLVisualizer: HDL Model Visualization with Simulation-Based Verification

Dominik Macko, Katarína Jelemenská
Slovak University of Technology
Faculty of Informatics and Information Technologies
Bratislava, Slovakia
macko@fiit.stuba.sk; jelemenska@fiit.stuba.sk

Abstract— **The usage of the HDLs (Hardware Description Languages) in a present digital system development process is indispensable. Although, their great contribution is undeniable, they also bring about several disadvantages. The textual form of a HDL model is less illustrative for a human being than schematic representation of its structure. Moreover, simulation of such models is most commonly displayed in a waveform representation, even though sufficient for verification, but hard-to-identify design errors, The paper presents a tool for supporting both, the model structure visualization and a proper form of the simulation results display.**

Keywords - digital system; hardware design; simulation; verification; VHDL; visualization

I. INTRODUCTION

With the VHDL (Very-High-Speed Integrated Circuits HDL) standardization in 1987, the massive development of supporting EDA (Electronic Design Automation) tools has begun. The HDL-based digital system design has many significant advantages – clearer design with fewer mistakes, verification by simulation, and technology-independence. However, the textual form of the structural HDL model is less illustrative for human being than a schematic representation, where the possible design errors are easier to detect. Consequently, many of the complex HDL design development environments support the conversion of an HDL model to its schematic representation. However, for simulation-based design verification a waveform representation of simulation results is the only one available in most of these tools. Although, this representation has a high verification power, especially the inexperienced designers find it hard-to-read and difficult to reveal the potential errors. This issue could be addressed by a tool that can display the simulation results directly in the schematic representation of an HDL model. Based on the analysis of available environments several possibilities of simulation flow visualization are offered. Connection color based logical value display is used in most of the logic circuit simulators (e.g. LOGiX [1]). Another possibility is to change a color of the port itself (used in TINA Design Suite [2]). These two possibilities work perfectly for one-bit signals, but when transmitting multi-bit signals one connection/port color cannot represent the logic value of multiple bits. In HDL simulator ModelSim [3] a connection label is used for displaying the signal value which is working well for multi-bit signals as well. However, the mentioned commercial tools are very complex and quite expensive to be used by beginners. Therefore, a simpler, intuitive and affordably priced tool is needed especially for education and novice designers.

II. PROPOSED TOOL DESIGN ASPECTS

These aspects are divided into two parts concerning the visualization of model structure and simulation results.

A. Intermediate Representation and Visualization

The VHDL model information is analyzed from source code using parser generator ANTLRv3 [4]. This information can be saved to an intermediate representation, suitable for visualization. An XML (eXtensible Markup Language) representation has the appropriate properties for this purpose since it can preserve the hierarchical structure, is widely used and allows easy to analyze contained information. The similar approaches were used in [5] and [6]. Four types of nodes have been defined in the proposed XML schema: *Architecture* (structure/behavior), *Port*, *Entity instance*, and *Connection*. The *architecture* in the XML file represents one hierarchical level. In fact, it is an entity instance at the upper hierarchical level. Its description includes the ports of the given entity instance (upper level ports), the entity instances of the given level, their ports, and the interconnections among the ports. To visualize the data it is necessary to design graphical representation of each object of the architecture. An *entity instance* is visualized as "black box", which functionality is not known. It is represented by a rectangle, displaying the entity name, the name of the respective instance, and the input and output ports. A *port* can be specified in different mode (in, out, inout, buffer, or linkage), according to which the shape of port is drawn. A *connection* is represented as a line, which interconnects two ports. In case of connection among the ports of internal instances, the signal name is displayed as well. The branching and splitting/joining of the connection has to be considered as well. For the architectures describing behavior of an entity instead of a structure, the visual difference has to be obvious. Therefore, in case an instance of the lowest level is selected, an object with no instance name and no entity name will be displayed. For structural model visualization a layout optimization algorithm is used that positions the entities in two

This is an accepted version of the published paper:
D. Macko and K. Jelemenská, "VHDLVisualizer: HDL model visualization with simulation-based verification," 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Tallinn, 2012, pp. 199-200.
doi: 10.1109/DDECS.2012.6219056
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6219056&isnumber=6219000

rows, sequentially, according to their occurrence in the VHDL source code. All the connections are positioned into a so called virtual bus located between the entities rows. Each connection has its own part of virtual bus reserved, therefore just the vertical parts of the connections are checked up for overlapping. If there is an overlap, the connection is shifted.

B. VHDL Model Simulation and Results Visualization

There are two ways to simulate the visualized VHDL model. The designed internal simulator allows the interactive simulation, where the designer can set up the input ports values for the selected hierarchical level. The values are spread through the design hierarchy, and subsequently the output ports values are calculated using the VHDL behavioral description loaded from the XML representation. When simulating test-bench entity, the input ports values are loaded also from XML. For the chosen simulation time, all the changes that should occur till that time are executed. The results of simulation are displayed directly in the schematic view. Another designed possibility is to simulate the VHDL model using an external simulator that will provide the simulation results to the VHDLVisualizer. The GHDL simulator [7] was chosen for this purpose. This simulator enables to check the VHDL code syntax, to analyze it and to run the simulation saving the simulation results into the standardized VCD (Value Change Dump) format [8] for which we created an analyzer. After the simulation is finished, the results are loaded from the generated temporary VCD file. Thus, for the visualized hierarchical level it is possible to display the value of the chosen port at the selected time. In the case lower level entity (not test-bench entity) is simulated using the external simulator, it is necessary to generate a new test-bench entity. In the architecture of this new entity, the values set up by the designer are assigned to the input ports (interactive simulation). Both simulators have limitations concerning the VHDL support, so their combination helps to increase this support. Based on the analysis of available solutions, we decided to display the simulation flow by means of connections labeling. Moreover, the color of the label changes when the signal state alters to improve the visibility. The simulation results are displayed in the labels located above the connections, but close to the ports (see Figure 1). In complex architectures, where a large amount of connections is displayed, this solution is much more convenient compared to the possibility to display the signal value somewhere in the middle of the connection. The simulation results can also be displayed in the commonly used waveform. For this purpose the external tool GTKWave [9] is used, loading the information from the VCD file generated by GHDL simulator [7].

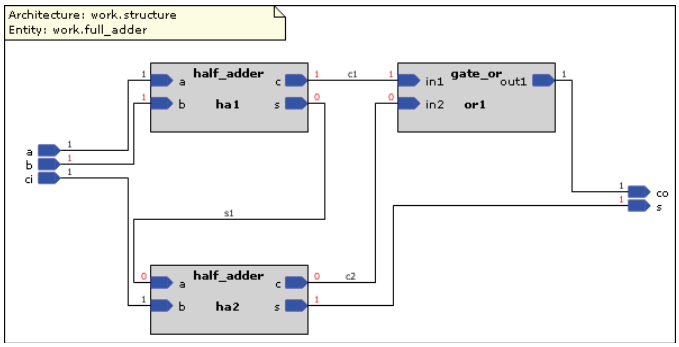

Figure 1. Structural VHDL model visualization with simulation results displayed

III. CONCLUSIONS AND FURTHER WORK

The paper is devoted to the problem of visualization and simulation of digital system models described in VHDL. We present the tool usable in digital systems design process. For a human being, the graphical representation of the structural model, generated by this tool, is better to understand and easier and faster to detect the errors made during the VHDL structural model creation. The VHDL model visualization is useful not only for verification purposes, but also for design documentation. For simulation results visualization in the schematic representation the solution possibility and algorithm were chosen and integrated into the VHDLVisualizer which brings the simplest, fastest, the most definite and the illustrative presentation of the simulation flow. Due to the unusual simulation and its visualization, this environment becomes a strong design and verification tool. The VHDLVisualizer nature makes it especially suitable for beginners in VHDL design - therefore education is the most natural area of application. However, it can become the useful verification tool for professional designs as well.

The tool is ready to be extended for other HDLs support such as Verilog and SystemC. Another possible extension is elimination of restrictions in the support of VHDL constructions, or object layout algorithm optimization. These extensions represent our further effort and future work.

ACKNOWLEDGMENT

This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 “Optimization of low-power design of digital and mixed integrated systems“).

REFERENCES

[1] CommTec - Software Engineering, “LOGiX - Simulation of logic circuits,” Simtel products, Online, December 2011. http://www.simtel.net/product/view/id/90288

[2] DesignSoft, “Analog, digital, symbolic, RF, VHDL, MCU and mixed-mode circuit simulation & PCB design,” Tina Design Suite, Online, December 2011. http://www.tina.com/English/tina/start.php

[3] Mentor Graphics, “ModelSim,” Mentor Graphics's products, Online, December 2011. http://www.mentor.com/products/fpga/simulation/modelsim

[4] R. M. Volkmann, “ANTLR 3,” Online, December 2011. http://jnb.ociweb.com/jnb/jnbJun2008.html.

[5] J. Petráš, “VHDL model visualization,” master theses, FIIT STU Bratislava (Slovakia), 2008, 85 p.

[6] M. H. Reshadi, B. Goji-Ara, Z. Navabi,”HDML: compiled VHDL in XML,” in VHDL International Users Forum Fall Workshop, Tehran Univ., 2000, pp. 69-74.

[7] T. Gingold, “GHDL – Where VHDL meets gcc,” Online, December 2011. http://ghdl.free.fr/

[8] IEEE, “IEEE standard Verilog hardware description language,” IEEE Standards (IEEE Std 1364-2001), September 2001.

[9] GTKWave Project, “Welcome to GTKWave,” sourceforge’s projects, Online, December 2011. http://gtkwave.sourceforge.net/