BHDL: Circuit design in B

Ammar Aljer, Philippe Devienne, Sophie Tison (LIFL)
Jean-Louis Boulanger (HDS)
Georges Mariano (INRETS)

LIFL: Laboratoire d'Informatique Fondamentale de Lille
HDS: Heudiasyc, Université de Compiègne
INRETS: Institut National de Recherche sur les Transports et leur Sécurité

Abstract. The main goal of this project is to provide a method of correct design of digital circuit. It combines the advantages of VHDL, the well-known language of circuit design, with the power of B method that guarantees the correct design (w.r.t. a formal specification). This allows avoiding the design test since it is "correct by proven construction". Furthermore, this project provides a tool, called BHDL, with a graphical interface for creating, editing, viewing and proving modular hardware architectures.

1 Introduction

The application of formal methods has been pursued since more that 10 years, but most of the investigations concentrate on simulation and verification. In this demo, we show in practice that it is possible to combine the well-known VHDL language and a formal method (B method) at a conceptual level, that is, from the very first steps of the digital circuit design. Since VHDL language and its tools are well adapted for hardware specification and B is devoted to formal software specification, this tool can be seen as the core of a co-design environment for rapid system prototyping. could take advantages from both approaches: VHDL point of view and sound mathematical B aspect .

2 Translating VHDL to B

VHDL supports hardware descriptions at various levels of abstraction as common normal programming languages. The circuits it describes are easily readable both as machines and as high level specifications. B method due to [7] is devoted to specification and development of computer software systems. Based on sound mathematical principles (set theory and first order logic), this method provides machine readability because of "abstract machine notation" and its modular approach. One constructs larger components from collections of smaller ones whose specification can be written little by little, by refinement. VHDL and B have enjoyed widespread acceptance as a standard to describe arbitrarily large designs. This tool uses these deep similarities to propose a translation from VHDL to B. The first work [2] has given the outlines of this translation from VHDL to B and one of the first applications was to translate the standard VHDL Std_logic_1164 package by hand [3]. The second step that we would like to demonstrate here takes advantage of the conceptual correspondences between B and VHDL for an automatic translation of the structural description for any VHDL program.

3 Development of BHDL tool

The subset of VHDL that we treat corresponds to VHDL-GUI [8], the graphical interface developed by Carl Hein (Lockheed Martin). VHDL-VGUI is a user-friendly interface for creating hierarchical diagrams, modules, boxes and their links. It provides an automatic translation to VHDL and Verilog. In collaboration with Carl Hein (VHDL-GUI), it has been extended for producing B code too. The grammar analysis has been made using ANTLR, a powerful compiler tool [9]. From a complete VHDL grammar, one may produce intermediate form trees (AST). By augmenting this grammar with tree operators, rewrite rules, and actions, one may produce an automatic translation at the highest level of abstraction. The correctness of the translation is easier to establish since these rewrite rules have a declarative semantic. Up to now, the B code is this exact translation of the VHDL one. Because of the last improvements of VHDL-GUI, we can now associate with the VHDL-GUI components (modules, boxes, links) formal specifications which are very useful for producing a richer B code that could not be expressed in VHDL (INVARIANT, PROPERTIES,INCLUDES, REFINES).

4 Features of BHDL

First of all,the user can design the circuit using a free graphical tool for capturing, drawing, editing, and navigating hierarchical block-diagrams, and for producing corresponding structural VHDL. He can create the circuit as black box with all of its connections with the environment (that is ENTITY in VHDL and abstract machine in B) . Also he declares the structure of the circuit with all of its contents (that is ARCHITECTURE in VHDL and implementation in B). A formal specification is associated with any box or module (that is, comments in VHDL and invariant, pre-conditions in B).

The translation from these diagrams to VHDL and to B is automatically exported because of ANTLR and its AST trees. More complete is the description of the diagram, more precise is its translation; the system can check from the first specification (the empty one) to the last one (the implementation in B) that each addition is compatible with the previous ones.

It is clear that BHDL tool can facilitate the developer's work: from one hand it enables him to use a graphical interface to design the circuit without directly using VHDL; from the other hand it automates the translation operation. Furthermore it enables to enrich VHDL entity description by a functional specification which is used in B model description.

References

[1] *BHDL web page* (in french) : www3.inrets.fr/B@INRETS/BHDL/bhdl/

[2] Boulanger JL, Mariano G, *Formal modeling of digital circuits using B method* (in French), Technical Report, INRETS-ESTAS, 1999.

[3] Aljer A, Boulanger JL, Devienne P, Mariano G, Tison S. *VHDL package Std Logic 1164 in B*, Technical Report, LIFL, 2000.

[4] Boulanger JL, Aljer A and Mariano G, *Formalization of digital circuits using the B method.* In CompRail VIII, Lemnos, Greece, June 2002.

[5] Boulanger JL, Aljer A and Mariano G, *B-HDL, an experiment to formalizing hardware by software formals specifications.* In EDCC4, Fourth European Dependable Computing Conference, France, October 2002.

[6] *Standard VHDL Reference Manual*, IEEE, 1993.

[7] Abrial JR, *The B Book - Assigning Programs to Meanings*, Cambridge University Press, 1996.

[8] *VHDL-GUI: Graphical User Interface for VHDL Modeling*, Ver 2.02, Jan 03 : www.atl.lmco.com/rassp/vgui/

[9] *ANTLR: ANother Tool for Language Recognition* : www.antlr.org

[10] *SAVANT: VHDL Analysis Tools* : www.ece.uc.edu/˜paw/savant/