B-HDL, an experiment to formalizing hardware by software formal specifications

Jean-Louis Boulanger[1], Georges Mariano[2], and Ammar Aljer[3]

[1] HEUDIASYC University Technic of Compiegne
jean-louis.boulanger@hds.utc.fr
[2] INRETS Georges.Mariano@.inrets.fr
[3] LIFL University of Sciences and Technology of Lille aljer@lifl.fr

Keywords : *B* method, formal development, software verification, safety-critical applications, VHDL.

Extended Summary

In this paper, we presented a part of our work to create *B* libraries which correspond to some *VHDL* packages, as the *STD_LOGIC_1164* package (see also [5]). This project enables us to take advantage of the power of the *B* method to develop a secure circuit. We write the specification of a desired circuit, then little by little we refine our specifications to reach to the implementation of this circuit which depends on the desired libraries. The *B* method due to J.R Abrial [2] is a formal method for the incremental development of specifications and their refinements down to an implementation. It is a model-based approach similar to **Z** [8] and **VDM** [6]. The software design in *B* starts from mathematical specifications. Little by little, through many refinement steps ([7]), the designer tries to obtain a complete and executable specification. This process must be monotonic, that is any refinement has to be proven coherent according to the previous steps of refinement. The *B* tool can automatically decide which induced proofs are necessary to verify this correctness. Then these proofs are produced either automatically for the simple ones or in cooperation with the designer for the comlex ones. The **abstract machine** is the basic element of a *B* development. It encapsulates some state data and offers some operations. In the *B* development, the proofs accompany the construction of software. Each time an abstract machine is defined or modified, there are proof obligations related to its mathematical consistency; if the machine is a refinement or an implementation, there are also proofs of its correctness with respect to the previous steps of the development chain. The *B* tool allow to generate automatically the proof obligations for each abstract machine. Generally speaking, the proof obligations will be all the more complex as concrete details are introduced. So, at the last refinement, the implementation, we obtain a secure software which does not need to be tested. On the other hand, *VHDL* (VHSIC - Very High Speed Integrated Circuits - Hardware Description Language)([1] and[3]) is an IEEE Standard since 1987. It is "a formal notation intended for use in all phases of the creation of electronic systems [...] it supports the development, verification, synthesis, and testing of

hardware designs, the communication of hardware design data ... " [1]. This presentation is devoted to showing the cross fertilisation between the circuit design methodology and the B method concepts. In [4], the famous standard VHDL package, the *STD_LOGIC_1164* , is transposed in the form of a B library as an example to be used as a set of B elementary components.In the same way, other VHDL packages can be translated as B circuit components in order to give to the designer a high-level view. Using this approach, one can develop a circuit of which each part of the specification is proven to be correct. A B circuit may be easily improved and it may be integrated with the other elements in the environment to satisfy safety conditions. We intend to create several libraries in B equivalent to the *VHDL* libraries, in order to facilitate the circuit design in B. Also this facilitates the transformation operation from B to *VHDL* . We try to find a common rule which may be used to automise the translation. Also we may solve this problem by creating a physical library in B that contains the characteristics of the basic electronic elements or by retranslating the results of a circuit development from B to *VHDL* . Semi-automatic translation of similar *VHDL* specification to B abstract machine is another way of work. In a first step, where we define the properties of the new compoents. The second step, is graphical, it introduce the components synthesis by composition of basic components.

References

1. *Standard VHDL Reference Manual.* IEEE, 1993.
2. Jean-Raymond Abrial. *The B Book - Assigning Programs to Meanings.* Cambridge University Press, August 1996.
3. R. Airiau, J.-M. Bergé, V. Olive, and J. Rouillard. *VHDL – Langage, modélisation, synthèse.* Collection Technique et scientifique des télécommunications. Presses Polytechniques et universitaires romandes, 1998.
4. Jean-Louis BOULANGER, Ammar Aljer, and Georges MARIANO. Concption sûr de circuit basée sur la notion de propriété. 14 ème journée internationales Génie Logiciel & ingénierie de systèmes et leurs applications, du 4 au 6 décembre 2001, ICSSEA 2001, 2001.
5. Jean-Louis BOULANGER and Georges MARIANO. Modélisation formelle de circuit numériques par la méthode *b*. Technical Report 1999-25-RT, 1999.
6. Cliff B. Jones. *Systematic Software Development Using VDM.* Prentice-Hall International, Englewood Cliffs, New Jersey, second edition, 1990. ISBN 0-13-880733-7.
7. C. Morgan. *Deriving programs from specifications.* Prentice Hall International, 1990.
8. J. M. Spivey. *The Z Notation: A Reference Manual.* Prentice Hall International Series in Computer Science, 2nd edition, 1992.

[1] Preface to the IEEE Standard *VHDL* Language Reference Manual.