

HIFSuite: Tools for HDL Code Manipulation*

Giuseppe Di Guglielmo, Franco Fummi, Graziano Pravadelli, Francesco Stefanni

{diguglielmo, fummi, pravadelli, stefanni}@sci.univr.it

Electronic Systems Design Group – University of Verona, Italy

<http://esd.sci.univr.it>

Abstract

The HIFSuite is a set of tools and APIs that provide support for modelling and verification of HW/SW systems by allowing manipulation and integration of HDL heterogeneous descriptions.

1. Introduction

The rapid development of modern embedded systems requires the use of flexible tools that allow designers and verification engineers to efficiently manipulate HDL descriptions throughout the design and verification steps. In fact, nowadays, it is common practice to define new systems by reusing previously developed components, that can be possibly modelled at different abstraction levels (TLM, RTL, etc.) by means of different hardware description languages (HDLs) like VHDL, SystemC, Verilog, etc.. Such an heterogeneity requires to either use co-simulation and co-verification techniques [1], or convert different HDL pieces of code into an homogeneous description [2]. However, co-simulation techniques slows down the overall simulation, while manual conversion from an HDL representation to another, as well as manual abstraction/refinement from an abstraction level to another, are not valuable solutions, since they are error-prone and time consuming activities. Thus, both co-simulation and manual refinement reduce the advantages provided by the adoption of a reuse-based design methodology.

To avoid such disadvantages in reusing already developed components and verifying their integration into new designs, we propose *HIFSuite*, i.e., a closely integrated set of tools and APIs that allow system designers to manipulate HW/SW descriptions in a uniform and efficient way.

2. HIFSuite Overview

The *HIFSuite* features are depicted in Figure 1. It allows designers and verification engineers to:

- analyze and parse VHDL or SystemC descriptions representing HW/SW systems;
- extract HDL Intermediate Format (HIF) representations of the parsed descriptions;
- manipulate and instrument the HIF representations by using already available HIF-based applications;
- define their own HIF-based manipulation tools by exploiting a powerful set of APIs;

- generate new VHDL or SystemC descriptions that reflect the changes introduced by the manipulation of the HIF representation;
- define their own conversion tools for supporting other hardware description languages.

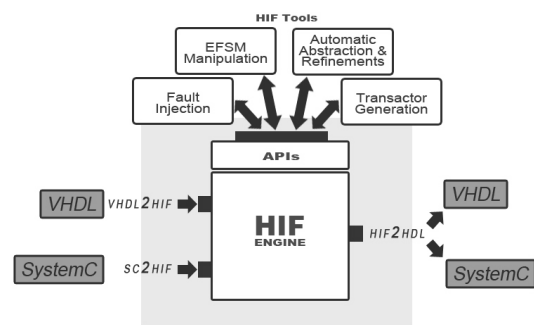


Figure 1: *HIFSuite* features.

The core of the suite is the HIF engine which is used for representing HDL constructs. The front-end tools *SC2HIF* and *VHDL2HIF* allow to convert, respectively, SystemC and VHDL descriptions into HIF models, while the back-end tool *HIF2HDL* allows to convert HIF models into VHDL or SystemC descriptions. The features of such tools are briefly summarized in Section 4.

Once the HIF representation has been obtained, a powerful set of APIs can be used for visiting and manipulating the HIF code. Currently, four tools have been developed by exploiting such APIs: a fault injector, an extended finite state machine (EFSM) manipulator, an abstraction/refinement tool, and a transactor generator. The feature of such tools are briefly summarized in Section 5.

3. HDL Intermediate Format

HIF stands for HDL Intermediate Format. It is an HW/SW description language structured as a tree of objects, similarly to XML. Each object describes a specific functionality or component that is typically provided by HDL languages like VHDL and SystemC.

Figure 2 shows an example of HIF code representing the entity of a simple design. Even if HIF is quite intuitive to be read and manually written, it is not intended to be used for directly describing HW/SW systems. Indeed, it is intended to provide designers with a convenient way for manipulating HW/SW descriptions as reported in Figure 1.

* This work has been partially supported by the European project VERTIGO FP6-2005-IST-5-033709.

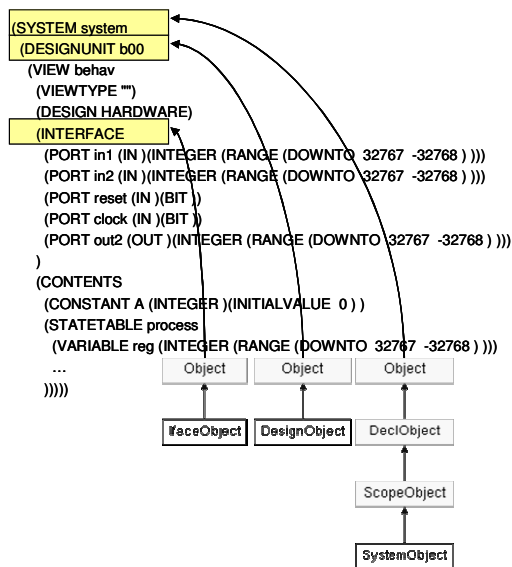


Figure 2: Example of HIF code.

4. Conversion Tools

HIFSuite front-end and back-end tools allow designers to convert HDL descriptions as described in Table 1.

Allowed HDL Translations		
	VHDL	SystemC
VHDL	X	X
SystemC RTL	X	X
SystemC TLM		X

Table 1: Conversion currently supported by *HIFSuite*.

Synthesizable RTL constructs are fully supported as well as the main TLM constructs. For the complete list of supported constructs please refer to [3].

5. Manipulation Tools

Currently, four HIF-based tools have been developed by exploiting the HIF library:

- A *fault injector*. It is used for perturbing HW/SW descriptions according to a selected fault model. Fault injection is a fundamental steps for fault coverage-based automatic test pattern generation [4].
- An *EFSM manipulator*. It is used for extracting extended finite state machines from HDL descriptions. Such a tool generates EFSMs that are easy-to-be traversed when verification techniques are applied that require to explore the state space of the design under verification [5].
- An *abstraction/refinement tool*, for semi-automatically abstracting RTL description towards TLM levels, and vice versa. Both the

abstraction and refinement methodology relies on manipulation of EFSM models [6].

- A *transactor generator*. It automatically generates transactors for allowing RTL-TLM mixed co-simulation and co-verification. It is used when components modelled at different abstraction levels are required to be integrated for rapid prototyping, before the refinement/abstraction methodology is applied to generate an homogeneous description [7].

6. Conclusion

In this paper, we presented an overview of *HIF Suite*, a set of tools and APIs that provide designers and verification engineers with the following advantages:

- The HIF library engine is written in C++, thus it is easy to be integrated into other programs.
- The HIF language is structured like a syntax tree, thus it is easy to write algorithms that manipulate the nodes of the tree.
- Systems described partially in VHDL and partially in SystemC, can be translated into the HIF representation, and then merged to obtain a unique final description in SystemC or in VHDL.
- The HIF library engine is structured to be easily extended. A special HIF object, called *PropertyObject*, is provided to describe non-standard or new features of other HIF objects.
- Tools, which manipulate HIF, can be used into workflows that adopt different HDL languages, regardless which these HDL languages are.

7. References

- [1] M. Bombana, and F. Bruschi, "SystemC-VHDL co-simulation and synthesis in the HW domain", *In Proc. of IEEE DATE*, pp. 106-111, 2003.
- [2] C. Cote, and Z. Zilic, "Automated SystemC to VHDL translation in hardware/software codesign", *In Proc. of IEEE ICECS*, pp. 717-720, 2002.
- [3] http://www.edalab.it/site/products_pages/HIFSuite.php
- [4] A. Fin, F. Fummi, and G. Pravadelli, "AMLETO: A Multi-language Environment for Functional Test Generation", *In Proc. of IEEE ITC*, pp. 821-829, 2001.
- [5] G. Di Guglielmo, F. Fummi, C. Marconcini and G. Pravadelli "EFSM Manipulation to Increase High-Level ATPG", *In Proc. of IEEE ISQED*, pp. 57-62, 2006.
- [6] N. Bombieri, F. Fummi, and G. Pravadelli, "A Methodology for Abstracting RTL Designs into TL Descriptions", *In Proc. of ACM/IEEE MEMOCODE*, pp. 103-112, 2006.
- [7] N. Bombieri, and F. Fummi, "Automatic Transactor Generation in TLM by Exploiting EFSMs", *In Proc. of DVCon*, 2007.

Acknowledgement

The authors would like to thank all the members of the ESD group of the University of Verona for their contribution in the development of *HIFSuite*.