

Static Analysis Tools for Soft-Core Reviews and Audits

Serafín Olcoz, Ana Castellví, María García, José-Angel Gómez

SIDSA

Ronda de Poniente 8, 2 A.

28760 Tres Cantos (Madrid). Spain

Abstract

Three navigation tools are presented to statically and interactively analyze Soft-Cores described in VHDL, [1]. These tools ease the adoption of mechanisms to perform reviews and audits procedures similar to those adopted in software development, [2]. These navigation tools help to better understand and reuse VHDL Soft-Cores. The three navigators are integrated in a VHDL-ICE environment, [3], to get design data management support.

1. Introduction.

According to the situation analysis provided by VSI Alliance¹, as semiconductor technology advances, the business pressure to design large ICs in a short time increases. Design reuse is expected to be a prevalent method for improving design efficiency of large ICs. In many cases, the reused blocks or cores are internally developed, but not always. Consequently, it is becoming critical for companies to increase their access to a variety of functional blocks, called Virtual Component (VC). VCs can be three forms: Soft, Firm, or Hard. Soft VCs are delivered in the form of behavioral or synthesizable Hardware Description Language (HDL), and have the advantage of being more flexible and the disadvantage of not being as predictable in terms of performance. Soft VCs typically have increased Intellectual Property (IP) protection risks because source code is required by the integrator.

The adoption of review and audit procedures coming from the software development world enhance reusability, and capitalizing on the IP library components investment.

Next section presents three new navigation tools² to support the adoption of Soft-Cores review and audit procedures. These interactive static analysis tools are fully and transparently integrated in the VHDL-ICE environment³ to ease their use without taking care of particular handling of the corresponding design data. Finally, section 3 presents the conclusions and future work.

2. Static Analysis Tools.

Static analysis methods include anything that can be done by formal (with mathematical bases) or informal analysis of the HDL description of a Soft-Core in conjunction with its processing steps (analysis, elaboration and simulation model creation), but not during them or while the simulation time advances. In order to reduce the scope of the possible analysis methods to be applied we have started by considering interactive navigation on the source code of the VHDL Design Units (DUs), the hierarchical components structure of a given Design Entity (DE) and its corresponding underlying simulatable model. These tools are specially useful when documentation and comments is not enough to properly understand and reuse the HDL descriptions of a given Soft-Core.

2.1 VHDL Design Unit Navigation.

This tool works on DE Declarations, Architecture Bodies, Package Declarations and Package Bodies. Its navigation capabilities are complementary to the source code edition. Its use is particularly indicated while developing a new Soft-Core, and specially when acquiring one that it is not very well documented, or their in-line comments are not enough to understand the functionality or the usage of the IP.

¹ VSI Alliance stands for Virtual Socket Interface Alliance. This alliance was formed from a common understanding of a looming bottleneck for the continued rapid evolution of the electronics industry and a shared vision for its solutions based on the continued use or “reuse” of existing functions, but which are designed in such a way as to make possible the mix-and-match of such functions from different sources onto a single silicon solution. See <http://www.vsi.org>.

² These tools are part of the VHDL Simulation Assessment tools under development in OMI-ESPRIT Project REQUEST. See <http://www.omimo.be>.

³ VHDL-ICE is a leading VHDL Integrated Common Environment, for UNIX and Windows NT, under development in the OMI-ESPRIT Project TOMI. See <http://www.omimo.be>.

