Chapter 7

MINING METADATA FROM SYSTEMC IP LIBRARY

Deepak A. Mathaikutty and Sandeep K. Shukla
FERMAT Lab, Virginia Tech, Blacksburg, USA
mathaikutty@vt.edu; shukla@vt.edu

Abstract
Exploring the design space when constructing a system is vital to realize a well-performing design. Design complexity has made building high-level system models to explore the design space an essential but time-consuming and tedious part of the system design. Reduction in design time and acceleration of design exploration can be provided through reusing IP-cores to construct system models. As a result, it is common to have high-level SoC design flow based on IP library promoting reuse. However, the success of these would be dependent on how introspection and reflection capability is provided as well as the interoperability standard are defined. This leads to the important question of what kind of metadata on these IPs must be available to allow CAD tools to effectively maneuver these designs as well as allows for a seamless integration and exchange flow between tools and design methodologies. In this chapter, we describe our tool and methodology that allow introspection of SystemC design, so that the extracted metadata enables IP composition. We discuss the issues related to extraction of metadata from IPs specified in an expressive language such as C++ and show how our methodology combines C++ and XML parsers and data structures to achieve the above.

Keywords Metadata, metamodeling, component composition framework, IP composition, reflection, introspection and interoperability

1. Introduction

Metadata is defined as “data about data” and is the kind of information that describes the characteristics of a program or a model. Information ranging from the structure of a program in terms of the objects contained, their attributes,
methods, and properties describing data-handling details can be exemplified as metadata. This class of information is necessary for CAD tools to manipulate the intellectual properties (IPs) within system level design frameworks as well as to facilitate the exchange of IPs between tools and SoC design flows. For such exchange they express integration requirements, performance, and configuration capabilities.

We classify EDA related metadata into two broad categories: (i) interoperability metadata and (ii) introspection metadata. Interoperability metadata enables easier integration of semiconductor IP and IP tools as shown by the consortium named SPIRIT [9], which allows design flow integration by utilizing metadata exchanged in a design-language neutral format. Consider the SPIRIT enabled flow from the ARM RealView SoC Designer tool to core-Assembler, the Synopsys SoC environment in [6]. The transfer is illustrated with a ARM1176JZ-STM processor subsystem design and the metadata captured through SPIRIT include: RTL I/O signals, support for bus-interfaces, parametric configurations, abstraction levels, memory map or remap information, interconnect layout, etc. This information exported from the SoC Designer allows a seamless import of the ARM1176JZ-STM processor subsystem into the Synopsys coreAssembler. Therefore, interoperability metadata serves as a common standard for exchange of multivendor IPs between tools and design flows.

Introspective metadata on designs in system level design frameworks allow CAD tools and algorithms to manipulate the designs as well as capture interesting properties about them. These features are also useful for debuggers, profilers, type/object browsers, design analyzers, scheme generators, composition validation, type checking, compatibility checking, etc. Introspection is the ability of an executable system to query internal descriptions of itself through some reflective mechanism. The reflection mechanism exposes the structural and run-time characteristics of the system and stores it in a data structure. The data stored in this data structure is called the metadata.

Consider a component composition framework (CCF) such as MCF [3]. MCF employs introspective metadata reflected from SystemC IPs to analyze and select implementations that can be used to create an executable for the abstract specification. The metadata that would facilitate IP selection and composition includes: register-transfer level (RTL) ports & datatypes, transaction-level (TL) interface signatures, hierarchy information, polymorphic characterization, etc.

In order to reflect metadata from IPs, the important questions that need to be answered are: “what is the metadata of interest?” and “how accessible is it?”. The first question would depend on how the extracted information will be used. The second question depends on how the metadata is specified. If the information is given through annotations to the implementation such as comments and pragmas, then the reflective mechanism would easily extract these.