UT SystemC Studio

Reihaneh Saberi, Amir Masoud Gharehbaghi, Parisa Razaghi, Anahita Naghilou, and Zainalabedin Navabi

{saberi, amir, parisa, anahita}@cad.ece.ut.ac.ir, navabi@ece.neu.edu

School of Electrical and Computer Engineering – Faculty of Engineering - University of Tehran - Tehran – Iran

http://cad.ece.ut.ac.ir/

Abstract

UT SystemC Studio is a SystemC environment for conversion between VHDL/Verilog and SystemC, SystemC simulation, SystemC assertion-based verification and testbench generation.

1. Introduction

The advent of modern System-on-Chip (SoC) designs with hardware and software parts creates a wide range of requirements. These requirements force the designer to decide on several design factors at the early stages of the design process. One of the most important factors is the hardware/ software partitioning scheme.

Traditionally because of the lack of a unique language with the ability to specify hardware and software together, decisions on system partitioning were difficult to make, and at the same time, unreliable. With the advent of system design languages such as SystemC, the design problems somehow reduced.

SystemC is a library based on C++ language which allows the designer to co-develop hardware and software parts of a design. The designer can develop the high level abstraction of the system and gradually refine it to the final model of the system. The significant benefit of using SystemC is having a complete model of the system in each step of the refinement. This enables the designer to evaluate the system in each step and solve the design problems.

UT SystemC Studio is a SystemC environment for conversion between VHDL/Verilog and SystemC, SystemC simulation, SystemC assertion-based verification and testbench generation. Figure 1 shows different parts of our environment.

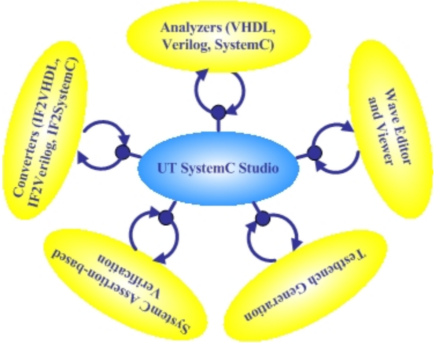
Analyzers (VHDL, Verilog, SystemC)
Wave Editor and Viewer
Testbench Generation
SystemC Assertion-based Verification
Converters (IF2VHDL, IF2Verilog, IF2SystemC)
UT SystemC Studio

Figure 1: The UT System Studio Environment

This paper is organized as follows. Section 2 describes our conversion process. Section 3 discusses our SystemC assertion library. Section 4 presents our GUI that includes a waveform viewer and an editor. Section 5 concludes the paper.

2. Conversion Process

The process of conversion between VHDL/Verilog and SystemC is done through analyzing the languages into an intermediate format and then converting it from the intermediate format into the output language. In this process two major categories of our environment are employed: analyzers and converters.

The output of any of the Analyzers is our intermediate format which is a set of classes that are designed based on VHDL [1], Verilog [2] and SystemC [3] languages. Each class of the intermediate format corresponds to a construct in a language. Similar constructs are mapped to one class. Figure 2 shows the interaction between analyzers, converters and the intermediate format.

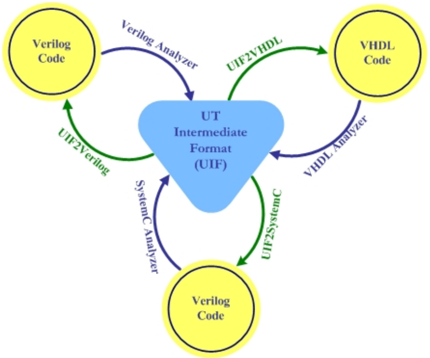

Figure 2: Interaction between analyzers and converters

Figure 3 shows conversion methodology and also the components of our environment involved in this task.

For generating the equivalent intermediate format of an input file, it is first parsed into a parse tree. After that, by traversing the generated parse tree the corresponding intermediate format is produced. In this step all syntax and semantic errors are reported by analyzer.

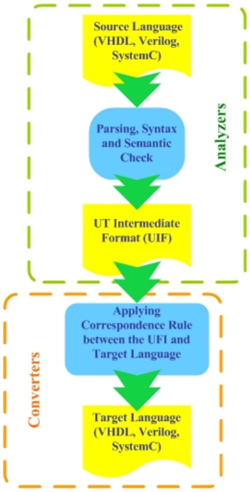
Source Language (VHDL, Verilog, SystemC)
Parsing, Syntax and Semantic Check
UT Intermediate Format (UIF)
Analyzers
Applying Correspondence Rule between the UFI and Target Language
Target Language (VHDL, Verilog, SystemC)
Converters

Figure 3: The steps involved in converting a source language to a target language

After the analysis phase, the intermediate format is converted to the target language using the converter for the target language. The converters work based on correspondence rules which are designed for mapping intermediate format classes to the target language construct.

The main advantages of VHDL/Verilog to SystemC conversion are:

- Enables reuse of previously designed VHDL/Verilog components in SystemC
- Accelerates System-Level design process
- Improves speed and quality of SOC design process

The main advantages of SystemC to VHDL/Verilog conversion are:

- Eliminates the tedious manual translation of RTL SystemC descriptions to supported entry formats of commercial synthesis tools
- Provides a reliable flow from SystemC descriptions to synthesizable HDL models
- Decreases time-to-market
- Provides a way for synthesizing SystemC descriptions, independent of synthesis tools

3. SystemC Assertions

Assertion-based verification is a powerful and popular verification methodology that is now used in many RTL hardware designs. Moving toward higher levels of abstraction and using system level languages such as SystemC requires a suitable verification methodology at that level. Addressing this issue, we have developed OVL [4] assertions with SystemC.

Each assertion is implemented as a SystemC module in two separate files (.cpp & .h). According to the number of OVL options in an assertion and for easier instantiation of these assertions, we have implemented several constructors for each assertion.

4. GUI

Our graphical user interface provides an environment for SystemC design and verification. It consists of the following parts:

1. Design manager: provides facilities for managing workspaces and projects.
2. Text editor: enables users to edit their VHDL-AMS, Verilog and SystemC source files. It has common edit features as well as syntax highlighting for all the languages.
3. SystemC simulation: is provided an interface and a link to the free OSCI library and Borland free C++ compiler.
4. Waveform viewer and editor: that is used for viewing results of simulation and also editing waveforms for testbench generation. The waveform viewer and editor support standard VCD format and its own binary format. The viewer has common features such as zoom, cursor, and changing visual signal attributes such as color and data format. The editor is used for testbench generation. It supports different data types of the languages and can be used to edit a single or repeated value in different ways. The editor can generate VHDL, Verilog and SystemC testbenches.

5. Conclusions

In this project, we have developed an environment for SystemC design and verification. It provides conversion from VHDL/Verilog to SystemC to reuse previously designed components and to speedup simulation. Conversion from SystemC to VHDL/Verilog is for SystemC synthesis, SystemC simulation, SystemC assertion-based verification, and testbench generation.

6. References

[1] IEEE Std 1076.1-1999, IEEE Standard VHDL Analog and Mixed-Signal Extensions, 1999.
[2] IEEE Std 1364-1995 IEEE Standard Verilog Hardware Description Language, 1995.
[3] IEEE Std 1666-2005 IEEE Standard SystemC Language Reference Manual, 2005.
[4] OVL assertions, available online at: www.accellera.org

Acknowledgement

The authors would like to thank the High-Tech Industries Center (HTIC) for their funding of this project. The support of HTIC has provided funding for personnel, equipment, exhibition, and management of this and several other CAD research and development projects at the University of Tehran.