UVM Standard-Compliant SystemC (AMS)-Based Verification Framework for Heterogeneous Systems

Yao LI, Zhi WANG, François PECHEUX, Marie-Minerve LOUËRAT,
Martin BARNASCONI, Thilo VÖRTLER, Karsten EINWICH
LIP6 Laboratory, University Pierre & Marie-Curie, Paris, France
NXP Semiconductors, Eindhoven, Netherlands
Fraunhofer IIS, Design Automation Division EAS, Dresden, Germany
{ Email: Yao.Li, Zhi.Wang, Francois.Pecheux, Marie-Minerve.Louerat @lip6.fr }

Motivation

Today's societal needs for innovative products in terms of communication, mobility, health, entertainment, and safety directly impact microelectronics design methodologies. The embedded systems are simultaneously software-driven, digitally assisted, complex and heterogeneous, and the quantity and nature of the interactions between its constituent parts grow substantially, but existing verification methodologies are mostly focused on pure digital devices and are completely decoupled from analog verification. Additionally, the existing methodologies tend to focus on implementation level verification. This presentation shows how the principles of the new UVM methodology can be soundly enhanced to offer to the test designer a flexible framework for the virtual prototyping of multi-discipline testbenches that supports both digital and Analog Mixed-Signal (AMS) at the architectural level.

1. UVM-Compliant SystemC (AMS)-Based Virtual Prototyping & Verification Methodology

Universal Verification Methodology (UVM) [1] is a convenient set of principles and techniques used to perform verification of Devices Under Test (DUT), with a specific focus on the reusability of the verification components and sequences. UVM-SystemC-AMS, developed by NXP, directly inherits from the SystemVerilog standard UVM and takes advantage of the SystemC[2] based virtual prototyping such as transaction-level modeling and the AMS[3] extensions for heterogeneous modeling.

In a typical UVM-SystemC-AMS scenario, the DUT is composed of digital and AMS parts, eventually running software. The DUT is connected to the rest of the testbench by means of the interface offered by SystemC and configuration mechanism offered by UVM. With the help of this mechanism, it is possible to dynamically connect the DUT pins to any pin-compatible UVM component at run-time.

A Universal Verification Component (UVC) is a reusable verification component in a **uvm_testbench** which consists of one or more **uvm_agents**. The **uvm_agent** is directly responsible for driving and monitoring the DUT. A **virtual_sequencer** is used to hand-over the test sequences to the **uvm_driver** through the sequencers. The **uvm_driver** handles the conversion of high-level transactions into bit-accurate (for digital) or analog signal samples (for AMS) that physically (i.e. at the electrical pin level) drive the DUT, the **uvm_monitor** is dedicated to the aggregation of samples/digital signals to compute performance indicators somehow related to the DUT output signal characteristics. Additionally, **uvm_monitor** may contains some analysis functionalities for coverage and checking. The role of the **scoreboard** is to determine whether the DUT operates correctly or not. It takes two streams of transactions (before and after DUT) as input and compare collected per−formances indicators values with the expected ones, coming from a golden model calculated with the stream of test sequences.

Constrained random verification generates stimuli (test scenario) for the DUT and the functional coverage that measures the percentages of the verification objectives that are met by the test plan. When the verification goals (coverage) is achieved, the simulation finishes.

2. Case study & verification process

The selected use case is a SPI-Controlled programmable filter which is composed of two parts: an Analog Mixed Signal (AMS) filter, and a digital SPI slave controller. The behavior of the filter is written in SystemC and its AMS extensions and controlled by a 4-bit word cmd_in. The signal cmd_in controls the gain and the bandwidth of the filter. The SPI slave controller is written in SystemC and has a genuine SPI interface, with a complementary reset input, a chip select input, a clock input and the 4-bit command data are shifted in using the mosi input.

In order to verify the DUT with UVM-SystemC-AMS, it has to be connected to 4 entities: a power supply generator, a wavefunc generator, a SPI driver that generates SPI-compliant commands and an analyzer in order to visualize and interpret the signals produced by the DUT filter. In standard UVM, the 4 entities are implemented as UVCs. Connections between UVCs and the DUT follow the interface configuration mechanism provided by UVM.

In our case, a virtual sequence consists in setting the power supply, resetting and writing a SPI command, and providing a frequency test request to generate a series of TDF samples with a specific operating frequency.

During the simulation, the 4 UVCs are carefully synchronized. The details of the stimuli propagation of different levels, the result reconstruction, the synchronization and the verification result will be shown in the presentation.

References

[1] Accellera Systems Initiative, Standard Universal Verification Methodology (UVM), http://www.accellera.org/downloads/standards/uvm/

[2] IEEE Computer Society, 1666-2005 IEEE Standard SystemC Language Reference Manual, IEEE, 1666-2005.

[3] Open SystemC Initiative (OSCI), Standard SystemC AMS extensions Language Reference Manual, www.systemc-ams.org