Testbenches for Advanced TLM Verification

Wolfgang Mueller
University of Paderborn/C-LAB
33102 Paderborn, Germany
wolfgang@acm.org

Wolfgang Ecker
Infineon Technologies AG
85579 Neubiberg, Germany
wolfgang.ecker@infineon.com

ABSTRACT
As verification activities dominate the systems development process efficient testbenches at an adequate level of abstraction are crucial factors to shorten time-to-market. We give an overview of the current state of ESL testbenches and the corresponding articles of the special session show advances and future directions of IEEE-1600-2011 SystemC in the context of testbenches for Transaction Level Modeling (TLM) [5][6][12][15][16].

Categories and Subject Descriptors
J.6 [**Computer-Aided Engineering**]: Computer-aided Design

Keywords
SystemC, Testbenches, Verification

1. Testbenches for ESL and RTL Verification
Several years ago, the move from Register-Transfer Level (RTL) to Electronic System Level (ESL) design also pushed the development of new verification methodologies and testbench libraries from Cadence, Mentor, and Synopsys: Universal Reuse Methodology (URM) [4], Advanced Verification Methodology (AVM) [11], and Verification Methodology Manual (VMM) [2]. They merged into the Open Verification Methodology (OVM) [14] and finally into the Universal Verification Methodology (UVM) [1]. UVM/OVM basically introduces extensible standard classes for structured implementation of RTL centric testbenches like (test) *environments*, *drivers*, *agents*, *sequences*, *monitors*, and *scoreboards* where transaction modeling (TLM) oriented styles are used for the testbench internal communication and for reference models (scoreboards). Complementary, to the previous methodologies, approaches for test specification [6] and testbench quality measures are developed. The most prominent approaches of latter are based on mutation based testing where also TLM testbench generation is under investigation [15].

Going in hand with the evolution of verification methodologies, several high level verification languages came up like the Property Specification Language (PSL) [8], the e language [7], and SystemVerilog [9] where the latter is a combined design, verification, and property language. All of them support (constrained) random test stimulus generation and a functional coverage metrics. At the same time, SystemC [10] became popular and well accepted as an ESL modeling and fast simulation language for mixed HW/SW virtual prototyping environments. With SystemC, the TLM 2.0 (Transaction Level Modeling) standard became accepted in industrial platform based design flows [5] and became finally combined with the most recent SystemC IEEE Std 1666-2011 [10]. TLM designs typically simulate 100x – 1000x faster than RTL models and thus enable early simulation and validation of the complete system including software. Most recent studies based on the application of GPUs give promising results to even faster simulations [16].

However, SystemC lacks verification features which were partly supported by SystemC libraries like the SystemC Verification Library (SCV) [13]. Additionally, RTL centric verification methodologies and languages can be linked to SystemC models via EDA tool specific transaction level interfaces. However, a SystemC implementation of a very small OVM subset came up [3], which was most recently extended towards functional coverage and constrained random stimulus generation and enhanced/migrated towards most relevant features of UVM for ESL verification [12]. Nevertheless, several points such as time quantum for coverage collection, time simultaneity of transactions, or reference models for TLM remain to be solved.

CODES+ISSS'12, October 7–12, 2012, Tampere, Finland.
ACM 978-1-4503-1426-8/12/09.

REFERENCES
[1] Accellera Organization Inc. Universal Verification Methodology (UVM), May 2012. www.uvmworld.org.

[2] J. Bergeron et al. Verification Methodology Manual for SystemVerilog. Springer, 2008.

[3] Cadence Design Systems, Inc. OVM-SC Library Reference Version 2.0.1, February, 2009.

[4] Cadence Design Systems, Inc. Universal Reuse Methodology (URM). www.cadence.com.

[5] W. Ecker et al. SystemC as A Completing Pillar in Industrial UVM Based Verification Environments. CODES+ISSS 2012, Tampere, FL, 2012.

[6] G. Di Guglielmo, G. Pravadelli. A Testbench Specification Language for SystemC Verification. CODES+ISSS 2012, Tampere, FL, 2012.

[7] IEEE Computer Society. Standard for the Functional Verification Language e. IEEE Std 1647-2011, 2011.

[8] IEEE Computer Society. IEEE Standard for Property Specification Language (PSL) - IEEE Std 1850-2010, 2010.

[9] IEEE Computer Society. IEEE Standard for SystemVerilog - Unified Hardware Design, Specification, and Verification Language - IEEE Std 1800-2009, 2009.

[10] IEEE Computer Society. Standard SystemC Language Reference Manual. IEEE Std 1666-2011, 2011.

[11] Mentor Graphics. Advanced Verification Methodology (AVM). www.mentor.com.

[12] M. F.S. Oliviera et al. The System Verification Methodology for Advanced TLM Verification. CODES+ISSS 2012, Tampere, FL, 2012.

[13] Open SystemC Initiative. SystemC Verification Library v1.0p2, 2006.

[14] OVM Homepage. www.ovmworld.org.

[15] M. Sousa, A. Sen. Generation of TLM Testbenches Using Mutation Testing. CODES+ISSS 2012, Tampere, FL, 2012.

[16] S. Vinco et al. SystemC Simulation on GP-GPUs: CUDA vs. OpenGL. CODES+ISSS 2012, Tampere, FL, 2012.