ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

A Case Study on Co-Simulation Co-Emulation Environments based on System C & System Verilog

Vrushali Bhanudas Kad

Assistant Professor, Dept. of I.T., Karmaveer Bhaurao Patil College, Vashi Navi Mumbai, Maharashtra, India

ABSTRACT: The flow of universal system-level design methodology consists of system specification, system-level hardware/software partitioning, co-design, co-verification using virtual or physical prototype, and system integration. In this paper, hardware part and software part of a design are described with SystemVerilog and SystemC, respectively after hardware/software partitioning. The functional interaction between hardware part and software part of a design is verified through co-simulation in pure software domain, and then verified through co-emulation after implementation of hardware part onto a specific hardware emulator. Verilog PLI provides a mechanism for Verilog simulators to invoke C functions which are registered as system functions through a complex process of library registration. Communication between software part and hardware part running concurrently needs IPC. Low-level system functions of device driver or kernel are called from C functions which are registered into Verilog PLI library. In contrast to Verilog PLI, SystemVerilog DPI which is used in co-simulation of this paper provides a way to interface with C/C++ or any other foreign language. Functions and tasks registered to the shared library using DPI can be called out like native ones. Model Sim recently supports SystemC simulation with built-in compiler for SystemC design unit, so the co-simulation of SystemC design units and SystemVerilog modules is carried out as one simulation process on Model Sim. After co-simulation, co-emulation using an FPGA- based prototype board is carried out.

KEYWORDS: co-simulation; co-emulation; SystemC; SystemVerilog

I. INTRODUCTION

Currently system-level design and functional verification methodology based on high-level abstraction becomes more important to increase the productivity of SoC design. In system-level design and verification, hardware/software partitioning of a system through design space exploration affects the structure and performance of the final system, and the importance of the verifying functional interaction between hardware part and software part is increasing ever[1]-[4].

SystemC [7][8][16] is a design language at multiple abstraction levels, and enables the design on a higher abstraction level to proceed to a synthesizable RT-level design through a progressive refinement .SystemVerilog[9]-[13] is a set of extensions to the Verilog HDL that allows higher level modelling and efficient verification flarge digital systems. SystemVerilog adds hardware functional verification constructs such as OOP (object-oriented programming), randomization, thread, etc.

Co-simulation[1]-[6] environments, which verify the functional interaction between hardware part and software part of the design, can be classified into homogeneous, heterogeneous, and semi-homogeneous in terms of hardware modelling language and software modelling language. As the functionality of hardware part becomes more complex and the scale of design becomes larger, the simulation performance decreases significantly in co-simulation. An alternative way is hardware/software co-emulation using real hardware such as FPGA for hardware part model.

Comparing the characteristics of a real hardware, emulator, and simulator on aspects of SoC design process such as speed, visibility, debug, setup time, coverage, or cost, a real hardware is fastest but has difficulty for verification. Simulator is easy for using but is hard to get significant performance, and emulator has better performance than simulator but has a disadvantage in cost.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

Most of previous research [2]-[6] about co-simulation have used IPC (inter process communication) with semaphore, pipe, socket, etc for communication between C program and HDL module. In this case, user-defined system functions are added to HDL simulator with Verilo PLI [13][14] in order for hardware module to use IPC, and more over system functions of device driver or kernel are called inside the added system functions. Because two processes, one for C program and the other for HDL module, are simulating on the base of kernel arbitration, the simulation performance decreases even though IPC provides portability and flexibility on co-simulation.

Co-simulation environment implemented in this paper is heterogeneous because hardware part and software part are designed with SystemVerilog and SystemC, respectively. The interface between SystemVerilog modules and SystemC design units is provided using SystemVerilog DPI (direct programming interface) [10]-[13][17]. Also, Model Sim recently supports SystemC simulation with built-in compiler for SystemC, so co- simulation of SystemC design units and SystemVerilog modules is carried out as one simulation process on Model Sim. In order to simulate SystemC design unit with Model Sim, the top-level design unit of SystemC should be modified using some macros provided by Model Sim[17]. Co-emulation is performed using iNCITETM,an FPGA- based hardware emulator, and iNSPIRETM, an integrated design environment released by Dynalith[2][18][19].

II. RELATED WORK

Several systems have been developed and built to analyze Accelerating system verilog Uvm based VIP to improve methodology for verificationof image signal Processing designs using HW emulator, for example Abhishek Jain et al.[19]who proposed a method Universal Verification Methodology (UVM) is a rich and capable class library that has evolved over several years from much experience with real verification projects large and small, and SystemVerilog itself is a large and complex language. As a result, although UVM offers a lot of powerful features for verification experts, it can present a daunting challenge to Verilog and VHDL designers who want to start benefitting from test bench reuse [20]. TestBench-Xpress (TBX) technology delivers the same functionality achievable in simulation with advanced and simple debug capabilities, but at 10-1000x of times faster performance. Additionally, it greatly increases verification productivity by using the same testbench for simulation and acceleration [21]. This paper describes development of Acceleratable UVCs from standard UVCs in System Verilog and their usage in UVM based Verification Environment of Image Signal Processing designs to increase run time performance.There are, however, a number of issues during verification of scan chain systems using RTL simulators. RTL simulation test time is dictated by the complexity of the test bench and the design, and the workstation CPU. Another factor is the RTL simulation tool used by the verification team. There are two types of RTL simulators used for verification: event-Based and cycle-based was presented by Bill Jason Tomas et al. [22].In this paper, was focus discussion on the use of event-based simulators. An event-based simulator updates whenever an event occurs, be it combinatorial or sequential. Event-based simulators can capture transitions through a combinatorial data path, which may not be aligned with a clock edge allowing users to capture issues such as glitches. Events are placed in a timing queue, which is evaluated in the order the events are placed. In terms of CPU usage, a majority of the workload is utilized to update the events queue. Since large SoC designs can contain thousands of registers, simulation time can take hours or even days depending on the system complexity. Also, a bulk of the transitions occurs between the combinatorial paths between all scan registers. To speed up this process, one may need to utilize a development platform so that the scan chain can be placed into hardware and removed from the simulation environment.

III. NATIVE-CODE CO-SIMULATION ENVIRONMENT

In system-level design methodology using SystemC, SystemC design units of hardware part can be automatically translated into RT-level HDL codes using translator such as Synthesizer provided by Forte Design Systems. In this paper, hardware modules are designed with SystemVerilog in person due to high license fee of translator.Native-code co-simulation environment does not use ISS (instruction set simulator). For reason of this, the application program to be executed in a processing core is not cross-compiled, but compiled using host computer compiler. Though the accuracy of simulation is a little poor, the simulation is performed efficiently and promptly because hardware resources controlled by application program are simulated.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

Co-simulation environment based on SystemC and SystemVerilog is shown in Fig. 1. It is a native-code co- simulation in which application program described with SystemC drives signals of System Verilog module through calling tasks those are defined and exported using DPIin SystemVerilog module.

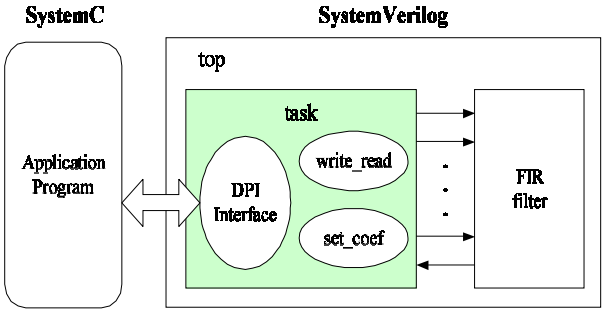

Figure1. Co-simulation environment of SystemC & SystemVerilog

A. Hardware Part Design

FIR filter, which issued in various applications regarding DSP, is designed with SystemVerilog as hardware module in this paper. The block diagram of the FIR filter intrans posed structure [15] is shown in Fig.2.

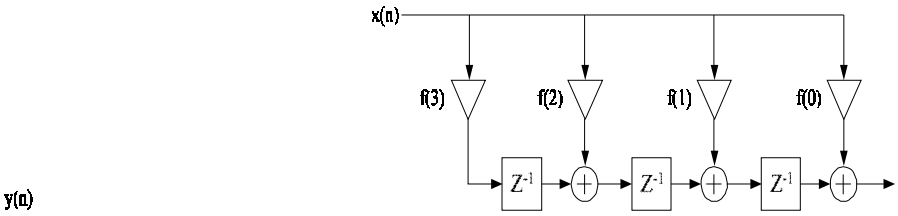

Figure2.Transposed FIR filter

As an extension of Verilog HDL, SystemVerilogh as characteristics of both hardware description languages and hardware verification language. System Verilog top-level module makes tasks such as write_ read and set_coef to be exported using SystemVerilog DPI in order for SystemC design unit to call it. Write read task does a job of driving input to FIR filter and dreading output from it, and set_coef task does a job of setting coefficients of FIR filter. The partial code of SystemVerilog top-level module is shown in Fig.3.

```
Module top;
Export "DPI-SC" task write_ read; export "DPI-SC" task set_coef; logic clk,reset;
 //...
 SC _ testbench SC_testbench ( ); FIR_SV FIR_SV ( //... );
Task write_read;
 Task set_coef;Input [7:0] data _in;
 output  [17:0] data _out;
//...
 End task

Input [7:0] coef1_in; input [7:0] coef2_in; input [7:0] coef3_in;
 input [7:0] coef4_in;
//...
 End task end module
```

Figure3.Partialcode of System Verilog top-level module

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

There are two instantiations; SC test bench is SystemC top- level design unit and FIR_SV is transposed FIR filter designed with SystemVerilog.

Exported tasks are automatically defined as C function prototype statement in sc_dpi header .h file when top-level SystemVerilog module is compiled with Model Sim Verilog module compiler, vlog. SystemC design unit has to contain sc_dpi header .h using #include pre-processor directive in order to call the exported tasks.

B. Software Part Design

In order to simulate SystemC design units which are calling the exported task from SystemVerilog with Model Sim, the following steps are required [17].

- contain sc_dpi header .h file using #include pre-processor directive
- replace SC_ main () with an SC_ MODULE, and identify the top-level SystemC module to Model Sim using the SC_ MODULE_ EXPORT macro
- compile SystemC modules using both sscom and MinGW gcc compiler in order to create shared library
- simulate the top-level design using vsim command The partial code of SystemC top-level design unit is shown in Fig. 4.

```
#include"sc_dpiheader.h"
SC_MODULE(SC_testbench){
svLogicVecValdata_in;
svLogicVecValactual_data_out;
svLogicVecValcoefficient1;
//...
voidproc_SC_testbench( );
sc_uint<18>ref_func_FIR(sc_uint<8>data_in);
SC_CTOR(SC_testbench) { SC_THREAD (proc_SC_testbench);  }
~SC_testbench ( ) { };
};

voidSC_testbench::proc_SC_testbench() {
svSetScope (svGetScopeFromName("top"));
//...
set_coef(&coefficient1,&coefficient2,&coefficient3,
&coefficient4);
while(true) {
//...
rand_value= (unsignedchar) rand();
data_in.aval=rand_value;
data_in.bval= 0;
write_read(&data_in,&actual_data_out);
expected_data_out=ref_func_FIR(data_in.aval);
//...
}
}
sc_uint<18>SC_testbench::ref_func_FIR(sc_uint<8>data_in) {
//...
expected=data_in*var_coef0+add1_delayed;
returnexpected;
}
SC_MODULE_EXPORT(SC_testbench);
```

The proc_ SC_ testbench () function registered as a process of the module using SC_ THRE A Dmacro, calls set_coef task to set coefficients of FIR filter. Also, it calls write_ read task with a generated random value to drive input to FIR filter and read output from it. Finally, it calls reference function, ref funcFIR ( ), to check the functional correctness of FIR filter.

Each variable that is passed through the DPI has two matching definitions; one for the SystemVerilog side, and one for the SystemC side. It is designer's responsibility to use compatibility pes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

IV. CO-EMULATION ENVIRONMENT USING FPGA-BASED PROTOTYPE BOARD

iNCITETM is a FPGA board for hardware/software co- emulation provided by Dynalith. iNCITETM provides a unique feature of hardware/software co-emulation through USB, where user design resides in the FPGA and testing program runs in the host computer. iNSPIRETM is an integrated design environment for FPGA-based emulation platform, iNCITETM [2][18][19]. The design flow using iNCITETMand iNSPIRETM is shown inFig.5.

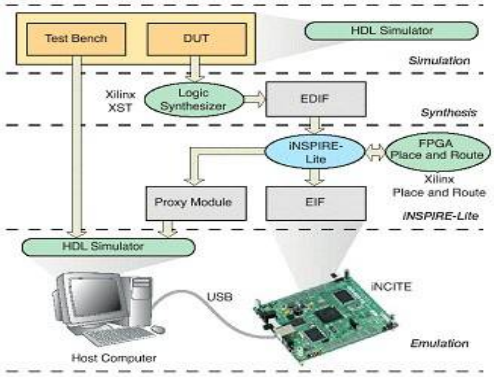

Figure5.The design flow using iNCITE^{TM} and iNSPIRE^{TM}

V. CONCLUSION AND FUTURE WORK

Co-simulation and co-emulation environments based on both SystemC and SystemVerilog after hardware/software partitioning of system-level design flow are described in this paper.

The typical communication based on IPC between C program and HDL module has disadvantages. In order for HDL simulator to use IPC, user-defined system functions have to be registered in Verilog PLI library. Moreover, the low-level system functions of kernel are called inside the registered system functions. These factors disturb an efficient co-simulation.

Co-simulation in this paper is performed with SystemVerilog DPI and Model Sim which recently supports SystemC simulation, so the co-simulation is efficiently carried out as one simulation process. And then co-emulation is performed using FPGA-based prototype board for hardware part implementation.
Co-simulation and co-emulation environments in this paper can be applied to co-verification.

REFERENCES

1. Jason R. Andrews," Co-Verification of Hardware and Software for ARM SoC Design", ElsevierInc., 2005.
2. Ando Ki, SoC Design and Verification: Methodologies and Environments, Hongreung Science, 2008
3. Yongjoo Kim, Kyuseok Kim, Youngsoo Shin, TaekyoonAhn,Wonyong Sung,KiyoungChoi,Soonhoi Ha, "An integrated hardware-software cosimulation environment for heterogeneoussystemsprototyping,"ASP- DAC, pp.101-106, 1995.
4. S. Chikada, S. Honda, H. Tomiyama, H. Takada, "CosimulationofITRON-basedembeddedsoftwarewithSystemC,"HLDVT,pp.71-76,
5. 2005.
6. S.Yoo,A.A.Jerraya, " Hardware/software cosimulation from interface perspective, Computers and Digital Techniques," IEE Proceedings, vol.152,issue 3,2005.
7. T. Jozawa, L. Huang, T. Sakai,S.Takeuchi,M. Kasslin, "Heterogeneous co-simulation with SDL and SystemC for protocol modelling," RWS, pp.603-606, 2006.
8. DavidC. Black, Jack Donovan, SystemC : From the Ground Up, EklecticAlly,Inc.,2004.
9. Thorsten Grotker,Stan Liao,Grant Martin,Stuart Swan, System Design with SystemC,KluwerAcademicPublishers,2002.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

10. Stuart Sutherland, Simon Davidmann and Peter Flake, SystemVerilogfor Design (2ndEdition): A Guide to Using SystemVerilog for Hardware Design and Modeling , Springer, 2006.
11. Chris Spear, SystemVerilog for Verification (2ndEdition):A Guide to Learning the Testbench Language Features, Springer,2008.
12. SystemVerilog3.1aLanguageReference Manual: Accellera's Extensions toVerilog, Accellera, Napa, California, 2004.
13. Stuart Sutherland, "SystemVerilog, ModelSim, and You," MentorUser2User, 2004.
14. Stuart Sutherland,"Integrating SystemC Models withVerilog and SystemC Verilog Models Using the SystemVerilog Direct Programming Interface," SNUG Boston, 2004.
15. Stuart Sutherland, The Verilog PLI Handbook: A Tutorial and Reference Manualon the VerilogProgramming Language Interface, KluwerAcademicPublishers, 2002.
16. U.Meyer-Baese, Digital Signal Processing with Field Programmable GateArrays, Springer, pp.79-81, 2001.
17. SystemCLanguageReferenceManual,http://www.systemc.org ModelSimSEUser's Manual,http://www.mentor.com
18. iNCITE User Manual, Dynalith Systems, http://www.dynalith.com[19] iNSPIRE User Manual, Dynalith Systems, http://www.dynalith.com [20] http://archi.cbnu.ac.kr
19. Abhishek Jain , Piyush Kumar Gupta, Dr.Hima Gupta and Sachish Dhar "Accelerating system verilogUvm based vip to improve methodology for verification of image signal Processing designs using hw emulator" International Journal of VLSI design & Communication Systems (VLSICS) Vol.4, No.6, December 2013
20. Rosenberg, S. and Meade, K., (2010) "A Practical Guide to Adopting the Universal Verification Methodology (UVM)", Cadence Design Systems, ISBN 978-0-578-05995-6.
21. Mentor Graphics Test Bench-Xpress user guide.Bill Jason Tomas, Yingtao Jiang and Mei Yang "Co-Emulation of Scan-Chain Based Designs Utilizing SCE-MI Infrastructure", International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 4, August 2014

BIOGRAPHY

Vrushali Bhanudas Kad is an Assistant Professor in Department of Information Technology, Karmaveer Bhaurao Patil College Vashi Navi Mumbai, MS, India. She received Master of Science in Information Technology (M.Sc.IT.) degree in 2012 from University of Mumbai, Mumbai, MS, India.Her research interests are Image Processing, Artificial Intelligence, Computer Networks, Pattern Recognition, Simulation Techniques etc.