A Study on Virtual Prototyping based Design Verification Methodology

Woojoo Kim, Kunhyuk Kang, and Seonil Brian Choi.
Design Technology Team, Infrastructure Design Center, S.LSI Division, Samsung Electronics,
1-1, Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do 18448, Korea

***Abstract*- Recently, design complexity and size of SOC is increasing exponentially however, SOC design time is decreasing on the contrary. To reduce the verification time of complicated SOC and improve the verification quality, this paper proposes a new verification methodology using the co-simulation technique of SystemC models on Virtual Prototyping (VP) and Verilog RTL (Register Transfer Level) model. Furthermore, in the proposed methodology, SW(Software)-based test cases developed through VP can be reused in RTL design/verification and SW validation. The test case reuse minimizes the verification planning time for each verification stage and enables the regression test from the beginning of IP (Intellectual property) design. Through this, it is possible to secure design quality early. Through the proposed verification methodology, not only functional verification but also specification verification can be performed at the same time. Therefore, design time increasing can be minimized by preventing design errors based on the same criteria for each design stage. In order to apply the proposed methodology into real SOC (System on a Chip) projects, there may be a time overhead for developing a SystemC model for all IPs. However, since VP is already being used for early SW development and performance estimation, many IP models were already developed, so the proposed verification methodology can show maximized TAT (Turn-around-Time) reduction effects without the time overhead for model development in real SOC projects. As a result, design verification time is reduced by 20%. Through test case reuse, verification planning time was minimized and regression could be started a month early by adopting VP co-simulation.**

I. Introduction

As design complexity increases, SOC architecture exploration or specification verification becomes more common process. To meet the customer's demands, it has been necessary to find optimal architecture by architecture exploration and to check whether the design specification is properly established or not. Furthermore, ESL (Electronic Semiconductor Level)-based VP is often performed to estimate functionality, performance, and power consumption early. Moreover, in automotive products, VP for SW development becomes essential process to develop SW in advance and secure safety early [1]. VP can be used for various verification platform by using traffic generators, SW, and direct SFR (Special Function Register) control from the test vectors used in RTL verifications. Test cases can be described as various languages such as SystemVerilog, C, C++, SystemC and so on, these are reused easily at the verification of each RTL, FPGA, and Silicon stage. Test cases in various languages are run on the SOC-level simple OS, and are configured as SFR sequences of HW (Hardware) or described and used as higher-level functions along with device drivers in the simple OS. In addition, in order to use the same verification cases at each stage, items for verification are managed from the IP design stage, and the verification results are version controlled with RTL models. Test case reuse not only reduces the test case description time because redundant activities can be avoided at each stage of verification, but also it guarantees the verification quality at each stage of development as the same level. It can be said that the method or direction is the same as the Portable Stimulus Standard (PSS). The test cases can be modified or added at verification each stage, but in most cases, those are able to be reused without any modification, which is effective in reducing the time to write the test case. In this study, the SystemC model and environment developed for Architecture Exploration were applied to RTL verification, and technical explanations for the proposed algorithms will be introduced in the following chapter.

II. Related Works

A. Virtual Prototyping

Through high-level programming languages such as SystemC/C++, the behavior and structure of IP can be modeled the same as the real design, thereby building a virtual SOC or system. Such a virtual system is called a virtual platform and can develop and verify HW [2] or SW[1] operations on a virtual platform. All of these activities are called virtual prototyping. When IP modeling is performed using SystemC, it is possible to make a difference in function/timing accuracy and model operation speed according to description precision. In order to verify the design and estimate performance or power consumption, the Cycle Accurate (CA) model or the Approximately Timed (AT) model, which is known to have a timing accuracy of around 90%, is generally used. These models have an operation

speed of several KHz to several MHz. For SW development, the operating speed of the model is very important, and the Loosely Timed (LT) model with 70~80% timing accuracy or the Function Accurate (FA) model that considers only the sequence without timing information is generally used [2]. The operating speed of this model is several MHz to several hundred MHz, which is widely used for early SW development or SW regression test in Automotive SOC projects. In fact, many semiconductor companies are using VP from the beginning of the project to perform architecture exploration based on HW performance and power estimation [2], and to complete SW development before silicon releasing [1].

B. SystemC-RTL Co-Simulation

The HW model described by SystemC and the HW model described by Verilog have the same specification, but the languages describing behavior and structure are different. The Verilog model has a signal-level interfaces, and the SystemC model has a signal-level or transaction-level interfaces. If a signal-level interfaces are used in the modeling, those can be connected without a special connecting models and can be simulated with the same CLK (clock) scheme. In case of SystemC models with TLM (Transaction Level Modeling) interfaces, those can be easily connected through transactors those can convert transaction-level to signal-level. In general, UVM (Universal Verification Methodology) widely used for RTL verification, is consist of TLM interfaces, and transaction conversion between TLM and Signal is easy [3]. For this reason, the SystemC or RTL Simulator basically provides co-simulation features. A typical SystemC-RTL Co-simulator consists of DUTs (Design Under Test) consisting of Verilog RTL and a test bench composed of SystemC. Simulation is often performed by transmitting the test sequence defined in the SystemC to the DUT and the I/O through the DPI (Direct Programming Interface) call. In this paper, the DUTs consists of both SystemC and RTL and Test Bench is written in SystemC. In some cases, SystemC models can include its own simulators. To run both RTL simulator and SystemC simulator, one of them shall be a master and the other shall be a slave. By the reason, each simulator must support both master mode and slave mode, but it is common not to support slave mode. To solve this problem, it is necessary to use a custom made DPI call or a custom wrappers. Alternatively, an old version of the simulator may be used.

C. Portable Test and Stimulus Standard (PSS)

In order to increase verification productivity, re-use of verification intent is required from IP level design to silicon level chip test stage. The concept of PSS is to only describe test intent and behavior once, and then target it for multiple abstraction layers, such as IP, block and SOC [4]. This includes the verification platforms simulation, emulation and silicon. These platforms have separate requirements and use different languages, as well as different approaches when testing aspects of the system [5]. PSS supports random and non-random data fields and structures that are familiar to SystemVerilog. From object-oriented programming they support familiar inheritance patterns. Figure 1 shows a representation of the anatomy of PSS, how one goes from a PSS description to executable code.

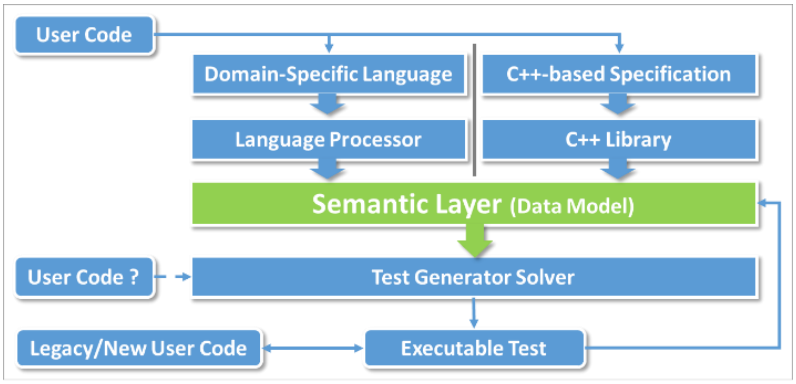

Figure 1. The anatomy of PSS [6].

III. THE PROPOSED METHODOLOGY

A. Proposed Design Flow

This paper proposed a verification methodology which uses VP for HW architecture exploration. And as the RTL design progressed, a method of verifying the function while replacing the SystemC model with the Verilog model was proposed. Figure 2 Shows the overall feature of the proposed methodology.

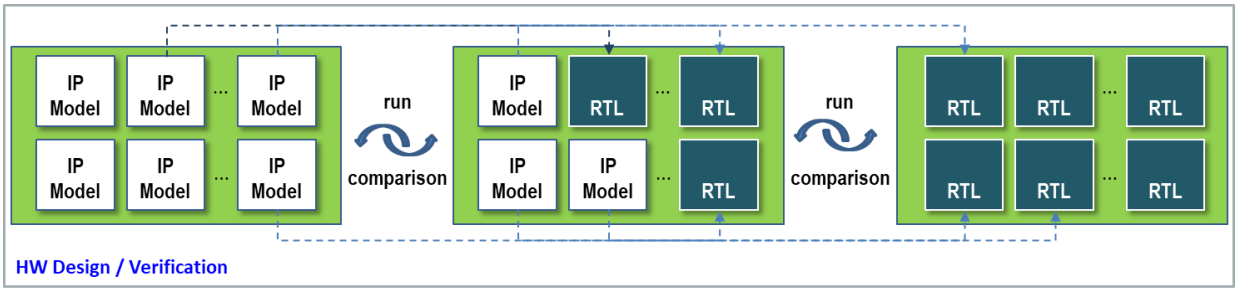

Figure 2. The Proposed Verification Methodology.

SystemC model development time is faster than Verilog model development time, and modeling is started during architecting, so it can be secured at the early design stage. The test cases for SystemC model functional verification can be described and be simulated using SWs or SFR sequences to RTL simulation, and the test cases are combined and used for the regression test. When the design starts, IPs are developed sequentially and delivered to the BLK or SOC development team. As shown in Figure 2, the SystemC model is replaced with the Verilog model and then verification is performed with the same test cases. Finally, when all IPs are replaced with Verilog models, it is the same as the existed RTL verification environment. Since the regression test is possible whenever each IP is replaced, the regression test can be started earlier than the existing RTL verification method. Therefore, the root cause of the error can be specified as the replaced IP and debugged quickly.

In general, the SystemC models created for early SW development may differ functionally and structurally from the actual RTL models. For this reason, even if the test cases are run, all functions defined in the RTL may not be verified in SystemC models. However, in this study, when developing the SystemC models, the behavior and structure were developed as same as the actual RTL design. This development method also requires systematic collaboration between teams so that all related teams can use the same.

B. SystemC – RTL Transaction

There are various methods for signal transaction between SystemC model and Verilog model. The Verilog model of RTL is based on cycle-driven and signal level transaction. It means that signal is transmitted at each cycle. On the other hand, the SystemC model has a more flexible interface structure. Signal level interfaces can be used, and TLM interfaces can be used for faster model operation. Figure 3 shows the difference between signal level transaction and TLM for read/write for AMBA (Advanced Microcontroller Bus Architecture from ARM) AXI.

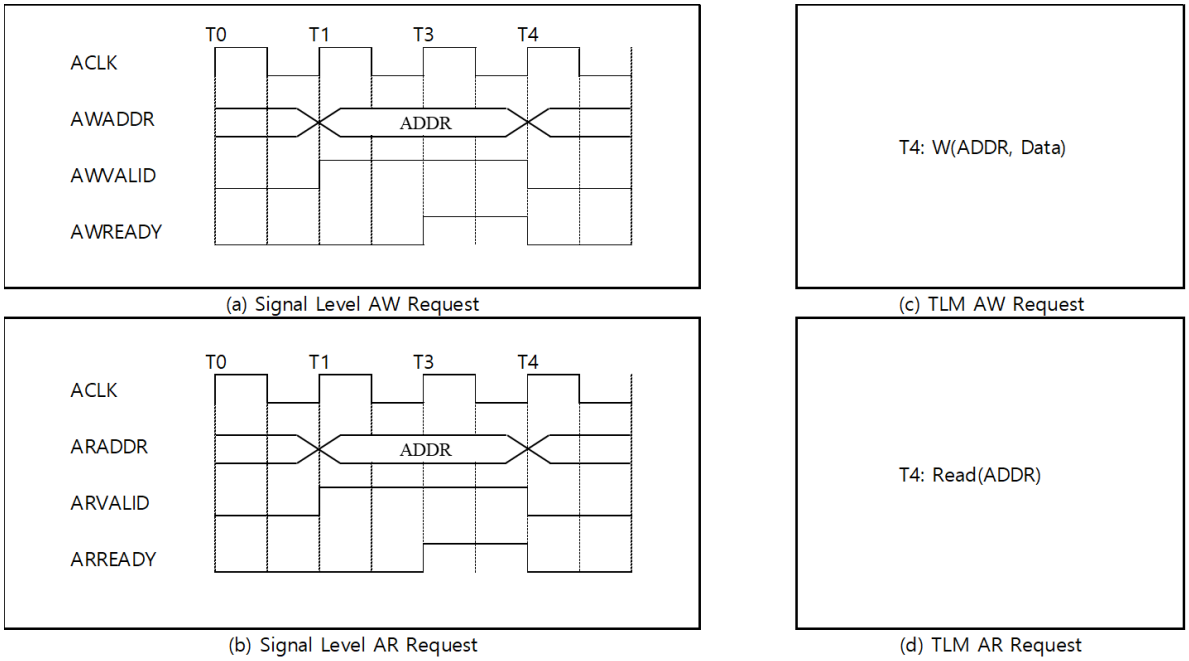

Figure 3. Format Comparison Between Signal and TLM.

For the SystemC models for performance estimation, cycle accurate model is used in general to calculate timing accurately even if the model operation speed is slow. In addition, in order to ensure the operation speed for SW development, Loosely-Timed TLM models which has slightly lower timing accuracy but a faster operation speed and uses only sequence orders, is widely used. In this paper, both the model developed for performance estimation and the model made for early SW development were used, so the transactor between the signal level interface of Verilog models and the TLM interface of SystemC models were developed together. Figure 4 shows the structure of the transactor used for RTL-SystemC Co-simulation in this paper. The TLM transaction data is separated into each AXI channels and each signals in the channel through the FSM (Finite State Machine), and signals are transmitted through READY/VALID handshaking. Conversely, the signals are translated and converted into TLM packets through the FSM, and transmitted to the TLM interfaces. The proposed transactor transmit data at 0 cycle and was written by SystemC/C++.

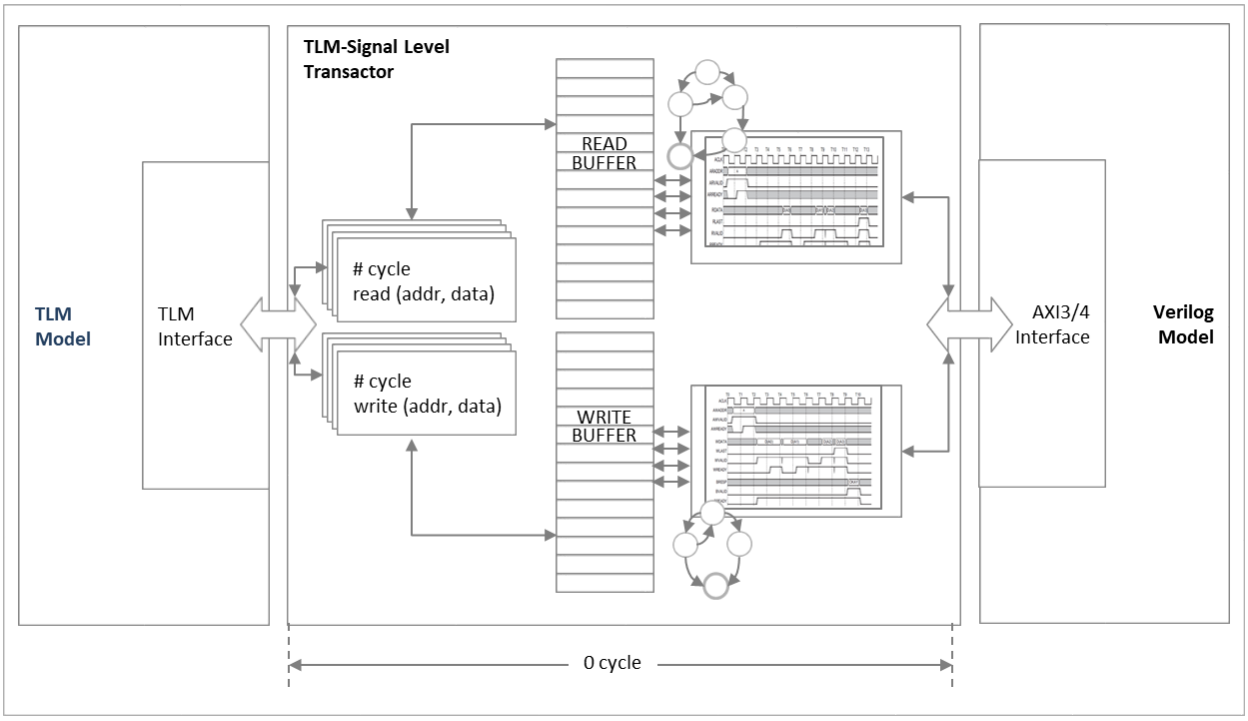

Figure 4. TLM / Signal Conversion Scheme.

C. Model Structure

In order to apply the simulation flow proposed in this paper, the same behavior and structure between the Verilog model and the SystemC model must be maintained. Also it is necessary to have the same functionalities, hierarchy, and interfaces. In this paper, SystemC modeling was made to include peripherals used in SOC based on RTL design hierarchies and functionalities. Although simplified than the real RTL design, the same operation can be guaranteed based on SFR control. Behaviors of IP are described first, then the behavior (or function) model is wrapped by SFR wrapper of them. In the case of SFR wrappers for each IP, a generator for parsing the SFR data sheet was developed in this study. Figure 5 shows the structure of each IP and the SystemC IP model is integrated into the Verilog TOP for co-simulation. A Register Information Signal Driver was developed for SFR communication with the Verilog modules, and an interface exists to correct some differences in hierarchy and to convert protocols. In Figure 5, the interface is shown as separated from the transactor to explain the function easily, but in the real implementation, the interfaces are included in each transactor as shown in Figure 4.

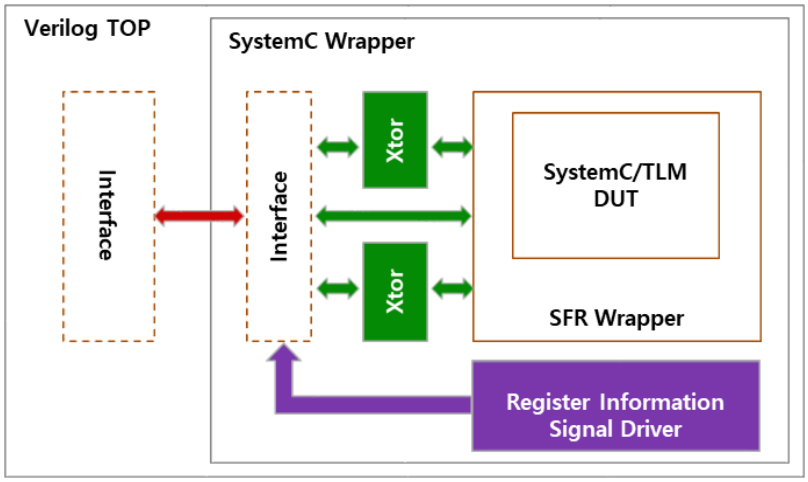

Figure 5. Transactor Architecture.

IV. Experimental Results

In this paper, the proposed verification methodology was applied to 100M G/C mobile flagship AP SOC to be confirmed. The proposed verification methodology was applied to the bus block and memory controller block, which are the most important for SOC performance. The Bus block developed the SystemC model for twelve IPs, and the memory controller developed the SystemC model for 7 IPs. A comparison was performed about the simulation speed and the verification period for the proposed Co-simulation method. The results of the experiment are shown in Table 1 and Table 2 as below.

Table 1. The Result of Simulation Speed Comparison.

	# of IP model	Simulation Speed [MHz]		
		SystemC	Co-sim	RTL
Bus	12	4.82	0.84	0.72
		x 6.69	**x 1.16**	-
Memory Controller	7	4.99	1.53	1.33
		x 3.75	**x 1.15**	-
Avg. Improvement Rate		**x 5.22**	**x 1.16**	-

Table 2. The Result of Verification Period Reduction.

	# of IP model	Verification Period [Month]		
		Existed	Proposed	Reduction Rate
Bus	12	8.1	6.4	**21.0%**
Memory Controller	7	8	6.35	**20.6%**

The existing verification method and the proposed method were compared using the same number of test cases. According to the simulation speed comparison result at Table 1, the SystemC only simulation was 5.22 times faster on average than the RTL only simulation, and the Co-simulation was 1.16 times faster on average than the RTL only simulation. In the case of Co-simulation, the speed may vary depending on various variables as well as the number and volume of RTL models, so the speed was measured whenever RTL models were included and the average speed of them was used for a comparison. As shown in Table 2, it was confirmed that the verification period was reduced by 21%. It is based on the verification completion date of SOC from the start of verification of IP and Block, and does not include the silicon validation period. Since the first RTL was released, a regression test for IP and block could be started, and verification began quickly, allowing bug finding and debugging from the earlier stage of the project. By the reason, debugging activities decreased during the SOC verification period or in the second half of the project, and in the end, it seems that the verification completion date has been pulled-in. According to the real design/verification flow, the verification scenario for each design stage is being developed in parallel, verification planning and test case development time reduction did not appear during the entire verification period, but it is expected that there will be benefits in terms of human resource reduction.

V. CONCLUSION

In this paper, a design verification methodology using VP was proposed. By applying the proposed methodology, the functional verification of each IP could be performed more quickly at the block level. And the regression test was started quickly for more than a month, and the same level of verification quality was secured during the verification period, which was reduced by 21% on average. In order to apply the proposed methodology in this study, SystemC modeling tasks for IP are required, which can be an overhead for resources such as human and time. For this reason, it may seem difficult to apply the proposed methodology. However, most semiconductor companies already use functional models for design widely, and many of them already use model-based design methodologies for early SW development, architecture development, performance, and power estimation. This allows the resource overhead to be hidden. Moreover, virtual prototyping is essential for the recently popular SW-driven HW development methodology, and it is seen as a good methodology to create synergy. The verification methodology proposed in this study was applied to specification and functional verification first, but it is expected that these kinds of verification can be performed quickly and easily by utilizing the performance and power estimation flow described above.

REFERENCES

[1] W. Kim, J. Lee, and S. B. Choi, "Early SW Bring-up Methodology using Full-SoC level Virtual Prototyping," Proc. DAC, Las Vegas, NV, June, 2019.

[2] T. Kogel. M. Doerper, T. Kempf, A. Wieferink, R. Leupers, G. Ascheid, and H. Meyr, "Virtual Architecture Mapping: A SystemC Based Methodology for Architectural Exploration of System-on-Chip Designs," Proc. SAMOS 2004: Computer Systems: Architectures, Modeling, and Simulation, Samos, Greece, July, 2004.

[3] Accellera Systems Initiative , Universal Verification Methodology (UVM), May 2012.

[4] Matthew Balance, Automating test from IP to SoC levels with portable stimulus. https://www.techdesignforums. com/practice/technique/automating-test-from-ip-to-soc-levels-with-portable-stimulus/, Accessed: 09.02.22.

[5] Gabe Moretti, Portable Stimulus: The Making of a Standard. Accellera, 2017. https://accellera.org/edps/hm/0247.html, Accessed: 09.02.22.

[6] Matthew Balance, "Making Legacy Portable with the Portable Stimulus Specification." Proc. DVCon, San Jose, CA, Feb., 2017.