# Seamless SoC Verification Using Virtual Platforms: An Industrial Case Study

Kyungsu Kang[1], Sangho Park, Byeongwook Bae, Jungyun Choi, SungGil Lee, Byunghoon Lee and Jong-Bae Lee
Design Technology Team, Memory Business, Samsung Electronics, Korea
[1]kyungsu.kang@gmail.com

***Abstract*— As SoC (System-on-Chip) complexity continues to increase, function/performance verification is required in the middle of design process (before tape-out) to reduce the possible risks ranging from over-design to non-compliance with the design specifications. In this paper, we propose a seamless SoC verification. The proposed methodology exploits a modern virtual platform (VP) technology which can combine high-level C++ firmware, timing-accurate SystemC models, and RTL (register-transfer level) designs. Thus, the full-chip level verification can be done at any design stages in the whole development process. With experimental results, this paper shows the benefits and lessons of using VPs.**

***Keywords*— *Virtual platform, Verification, Design space exploration, SystemC/TLM, Solid-state drive (SSD)***

## I. INTRODUCTION

In this paper, we propose a seamless SoC verification methodology which is a continuous verification process from design specifications to silicon available with full-chip level test scenarios and simulations. It helps engineers determine the design parameters at the early design stage and verify the gradually implemented HW/SW designs with a consistent verification environment until a hardware prototype is ready. This seamless verification is feasible thank to the support of co-simulation between SystemC models and RTL designs in VP technology. The contributions of this paper are as follows.

- This paper presents an industry case study of applying VP technology for the seamless SoC verification. As far as the authors know, this is the first work that uses VPs in the whole design and verification processes, not just in a part of design stages.
- This paper reports practical issues, strategies, and benefits of using VP technology for a real product, i.e., SSD (Solid State Drive), development in industries.

Section II presents the proposed verification methodology. The lessons learned in the case study are summarized in Section III.

## II. PROPOSED VERIFICATION METHODOLOGY

Fig. 1 shows the proposed SSD development process. In the early design stage, VP-1, which consists of only SystemC models, is used to determine product architectures as well as IP design parameters e.g., SRAM buffer size, data-flow control policy, operating clock frequency. Because VP-1 is much faster and flexible than RTL simulation, many full-chip level test scenarios can be simulated. However, the development of VP-1 is not just assembling existing library components (e.g., processor model, bus model, memory model, etc.) available in commercial tool libraries, but it demands significant amount of manual works, e.g., developing SystemC models for legacy IPs with a tight development schedule. For that, we should not model all the components of SSD with full cycle accuracy, but classify them with respect to the accuracy requirements, i.e., 1) full cycle accuracy, e.g., asynchronous buffers, 2) latency-throughput accuracy, e.g., DMA controllers, 3) untimed accuracy, e.g., HOST computers.

As time goes, IPs are implemented in order and they need to be fully verified with respect to the specifications. Usually, in development, interconnects and memory controllers (i.e., SRAM/DRAM controller) are implemented earlier than the other IPs, because IP designers get help from third-party design tools [1]. Verifications of them are crucial because the behavior of generated RTL designs from the design tools might not be what the designers expect and the performance bottleneck of them seriously degrades the whole SoC performance. By replacing some SystemC models in VP-1 with the target RTL DUTs (Design Under Test), VP-2 makes it possible to validate them with the same full-chip level scenarios used in VP-1 before the other master/slave designs are available. In VP-2, in order to make a communication between SystemC model and RTL DUTs, there must be a transaction-signal converter which connects a TLM interface of SystemC model to the corresponding signal inputs and outputs of RTL design. For the protocol converters, there are commercial tools that supports

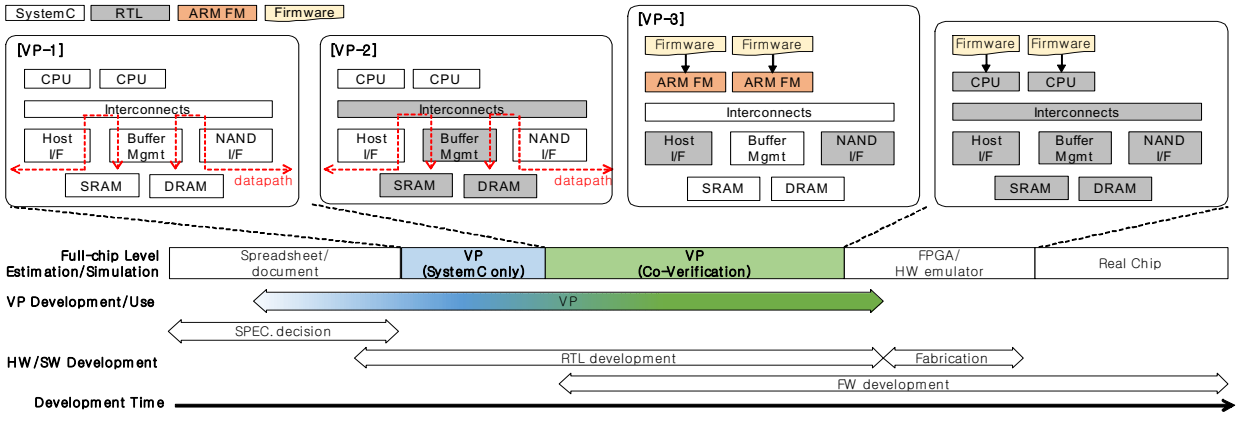


Figure 1: The proposed design process using VPs for seamless SoC verification

them as libraries [2]. Through VP-2, we basically validate the connectivity, latency, and throughput of interconnects as well as the efficiency, latency, and throughput of SRAM/DRAM. In addition, we evaluate various candidate designs that have different parameter configurations. In our experience, with various interconnects evaluations, we got the final interconnect design with the gate-counts and power consumption reduced by 41% and 42%, respectively. For the power estimation, there are commercial tools [3] that estimate the power consumption from simulation waveform dump.

VP-3 is used to verify implemented RTL designs that communicate with a CPU running firmware. In the conventional design process, since a set-up of hardware prototype (e.g., FPGA or HW emulator) happens much later, both HW and SW designers have no choice but to lean on only RTL simulation for the HW/SW verifications. Because of the limitations of RTL simulation, i.e., lower simulation speed, no support of SW debugger, difficulties to make full-chip level verification environment, it is hard to fully verify their implementations. In VP-3, a commercial virtual model, i.e., ARM fast model[4], is used in conjunction with SW debuggers (e.g., Trace-32 [5]). Thus, VP-3 helps not only reduce the IP verification efforts but also develop and verify early SW functions (e.g., boot code, basic operations, etc.) before the hardware prototype available. In our experience, VP-3 was available 6 weeks ahead of FPGA. It was used for early SW bringing up as well as design flaw detections. Thus, the set-up time of FPGA has been reduced by 2 months.

## III. EXPERIMENTAL RESULTS (LESSONS LEARNED)

Creating a VP for early design stage demanded modeling SystemC components within a given tight development schedule with the limited man-power. Thus, a practical strategy was required, and we did only the least of what we had to do. As development goes, we replaced SystemC models with the corresponding RTL designs. This approach, i.e., first creating a VP with fast and flexible models only, then replace them with slow-but-accurate ones later, makes it possible to do seamless SoC verification while meeting the time-to-market constraint for the real product development. Table I summaries the design faults being found from a SSD development project. In order to find the design faults, a long-term full-chip level simulation was required, where either spreadsheet program or RTL simulation is not possible. We could fix them before the hardware prototype available.

In SystemC-RTL co-simulation, the simulation speed was shackled by RTL simulator. However, in our case, it is endurable for verification (e.g., ~2.2 IOPS for normal SAS commands, ~0.3 IOPS for normal NVMe commands in the view of wall clock time). Table II shows the simulation time of VPs which is normalized with respect to RTL simulation time. For the simulation, HOST sends 32 commands for 4K SSD write. In Table II, Host I/F Sub-System includes Host I/F, Bufffer Mgmt, RTL memory models, and Interconnects, but not include CPUs. Because the simulation speeds of VPs are much faster than the RTL simulation, it extremely reduces verification effort as well as turn-around time (TAT).

Even though the use of VPs has many benefits as described above, there are some limitations. First, the current product design process does not consider VP development. The project deadline, man-power, as well as all the design documents were usually determined only for HW and SW engineers, not for VP modeling engineers. Also design specifications were frequently changed even during the design progress. To tackle the issues, it was required to contact each HW and SW engineer on a person-to-person basis, not just relying on the documents. Second, we used ARM Fast Models that are loosely-timed (LT) models. Because they are not cycle-accurate virtual models, it is hard to use them for SW optimization in the view of performance. Thus, we consider to use ARM cycle model [6] for the next project. At last, in this work, the VP was developed to endure only for the functional verifications of main data paths. If you want to use it for HW/SW regression test, you need to develop a different type of VPs, either a VP that consists of only fast virtual models or a hybrid emulation that makes it possible to communicate between virtual models and real designs synthesized in a hardware prototype.

TABLE I. DESIGN FAULT LIST DETECTED THROUGH VPS

| *Verification Category* | *VP Type* | *Count* |
|---|---|---|
| Interconnect & Buffer performance degradation | VP-2 | 3 |
| HW performance degradation | VP-1, VP-2, VP-3 | 7 |
| HW/SW functional fault | VP-2, VP-3 | 16 |

TABLE II. SIMULATION PERFORMANCE COMPARISONS

| *VP Type* | *Simulation Time* | *RTL IP* |
|---|---|---|
| VP-1 | 0.0015 | None |
| VP-2 | 0.0410 | Interconnects |
| VP-3 | 0.0761 | Host I/F only |
| RTL | 1.0000 | Host I/F Sub-System w/o CPUs |

## IV. CONCLUSION

In this paper, we presented an industry case study of applying VP technology to real product development. The use of proper VPs makes the seamless SoC verification possible. In the early design stage, VP was used to determine the performance-related design parameters. As development goes, SystemC models were gradually replaced by real RTL designs and the VP was used to verify them with full-chip SoC test scenarios. Product SW was brought up early on ARM fast model so that HW/SW functional verifications were done. In our experience, the VP for HW/SW co-verification was available 6 weeks ahead of FPGA and the use of VP reduced the set-up time of FPGA by 2 months. Through VPs, many design faults were detected, which usually might be detected a hardware prototype. The interconnect power consumption was reduced by 42%. Overall, the proposed methodology helped decrease the HW resources as well as the development time, thus it raises SSD's competitiveness in the world market.

## REFERENCES


[1] ARTERIS, FlexNoC, http://www.arteris.com/flexnoc

[2] SYNOPSYS, Platform Architect, http://www.synopsys.com/Prototyping/ArchitectureDesign

[3] ANSYS, PowerArtist, https://www.ansys.com/ko-kr/products/semiconductors/ansys-powerartist

[4] ARM, FastModels, https://developer.arm.com/products/ system-design/fast-models

[5] LAUTERBACH, Trace-32, http://www.lauterbach.com

[6] ARM, CycleModels, https://developer.arm.com/products/system-design/cycle-models