Versatile SoC Development Environment Supporting HW/SW Co-Design And Mixed Abstraction-level Simulation

Sang-Heon Lee† Heejun Shim† Ki-Yong Ahn† Seonpil Kim† Yun-Sik Woo‡ Chong-Min Kyung†

†Department of EECS
Korea Advanced Institute of Science and Technology
Daejeon, 305-701, Korea
E-mail: {shlee, shimy, ahnky, spkim, kyung}@vslab.kaist.ac.kr

‡Center for SoC Design Technonogy
Korea Advanced Institute of Science and Technology
Daejeon, 305-701, Korea
yswoo@vslab.kaist.ac.kr

***Abstract*— We will demonstrate two SoC design methodologies, verification environment based on hardware prototyping, named ProBase, and mixed abstraction-level simulation.**

I. PROBASE

ProBase is a flexible SoC development platform supporting hardware and software co-design of SoC. It is based on a prototyping technique similar to ARM Integrator®. But it has several important improvements, which distinguish it from ARM Integrator®. These include co-working with off-the-shelf simulation accelerator for ease of debugging, provision of simulation model of the platform itself for easy system integration, and addition of external I/O ports for general use. ProBase is also compatible to ARM Integrator®. So it can utilize all the benefits of it including large collection of ARM and DSP processor models.

For enhancing hardware debuggability, ProBase can co-work with off-the-shelf simulation accelerators to utilize powerful hardware debugging features of them. For that purpose, hardware components under test are modeled in the simulation accelerator. ProBase and simulation accelerator are interconnected with a bidirectional port where the AHB bus signals are packed. But the bidirectional port does not have sufficient number of pins to transfer all the signals of AHB. To remedy this problem, we introduced AMBA splitter, which hides the lack of pins by time multiplexing bus signals.

The simulation model of the platform makes complete the SoC design steps, finding out plain bugs before prototyping using real chip and FPGA. In the simulation model, software applications in core run in ISS for the core, and hardware designs run in HDL simulator such as ModelSim®. And the two heterogeneous simulators are connected via IPC channel.

II. MIXED ABSTRACTION-LEVEL SIMULATION

This demonstration shows cycle-accurate mixed abstraction-level simulation and simulation acceleration method. This enables us to utilize transaction-level test vectors which are usually already implemented in early design steps and also easy to generate than HDL test vectors, to verify RTL design. As there is no commercial simulation environment that can efficiently handle both transaction-level and RTL models at the same time, we employed two simulators for each abstraction-level modeling. Figure 1-(a) shows transaction-level modeling of an embedded system, while figure 1-(b) shows mixed abstraction-level modeling of transaction-level and RTL designs. As shown in figure 1-(b), to bridge different abstraction levels of communication between the two simulators, transactors are implemented and inserted between them. Transactors translate transactions, such as request, grant, read and write, to signal-level information and vice versa. In the mixed-level simulation, simulation speed is limited by an RTL simulator which is much slower than a transaction-level simulator. To compensate the speed degradation of mixed-level simulation, we used a FPGA-based simulation accelerator instead of an RTL simulator, as shown in figure 1-(c). For the connection between a transaction-level simulator and a simulation accelerator, we implemented a hardware acceleration wrapper that accepts signal-level information from transactors and then calls the signal-level APIs of the simulation accelerator.

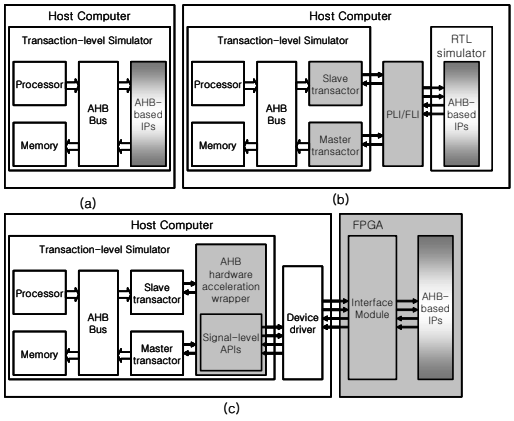

Fig. 1. (a) Transaction-level modeling of embedded system (b) Mixed abstraction-level modeling of transaction-level and RTL (c) Mixed abstraction-level modeling with simulation accelerator

We used MaxSim® from ARM as a transaction-level simulator and ModelSim® as an RTL simulator. For a simulation accelerator, iPROVE® from Dynalith was used.