Multi-processor SoC (MPSoC) design is much more difficult and may need much longer design cycle than conventional SoC design with one or two processors. This is mainly due to new design challenges such as parallel programming, mapping, debugging of concurrent processes running on different processors, etc. With this in mind, re-design of an MPSoC at a late design stage can be very costly, and more thorough validation in early design stages is crucial to reduce design cycle as well as design risk. Since system validation methods in early stage of design flow should have high availability and visibility, co-simulation in various level of system abstraction is widely used to validate the system. Especially, timed co-simulation is indispensable to validate complex MPSoC because the behavior of complex MPSoC is highly dependent on the timing of interactions among the system components. However the low performance of current timed co-simulation environments causes designers to have difficulties in whole system-level simulation with full target software including OS.

The performance bottleneck in timed co-simulation comes mainly from the fact that the simulation models should be synchronized at every timing granularity to validate timing behavior of the system. For example, the simulation models have to be synchronized at every clock cycle in cycle-accurate co-simulation, and at every delay boundary in delay-annotated cycle-approximate co-simulation. In the case of multiple-process co-simulation environment, the synchronization overhead becomes significant because the simulators exchange messages via Inter-Process Communication (IPC) for the synchronization. In the case of single-process co-simulation environment, the simulation kernel performs context switches between simulation models to synchronize them. Although the IPC overhead is removed during the synchronization in the single-process environment, still, the context switching overhead dominates the performance of cycle-approximate simulation.

To reduce the synchronization overhead, the thesis proposes Synchronization Time-point Prediction (STP) method, which consists of two phases: static analysis of simulated models and dynamic scheduling of synchronizations. In the static analysis phase before simulation, it estimates minimum execution time from every execution point to the nearest synchronization point in the model. Then, during simulation, STP synchronization is performed by pessimistically predicting the next synchronization time-points based on the estimates.

This thesis proposes a fast cycle-accurate co-simulation environment which utilizes the STP method. Multiple instruction set simulators and RTL hardware models are synchronized in the cycle-accurate co-simulation environment. The assembly codes of the target software and the RTL hardware models are analyzed statically for the STP method. According to the experiment, the speed of cycle-accurate co-simulation is improved up to 750K cycles per second.

A fast cycle-approximate co-simulation environment is also proposed in this thesis. Delay annotated software codes and transaction level models of hardware are co-simulated in the cycle-approximate co-simulation. The techniques to analyze such abstract models and schedule minimal number synchronizations during cycle-approximate co-simulation are presented. Experiments show that the approach achieves the simulation speed up to 83M cycle per second in cycle-approximate co-simulation.

* Note: The text above is the abstract of the thesis.
You can use the full-text by clicking the "Link" (URL) of item record and the viewer program will be installed automatically. In case of having problem with the viewer program installation, please download it manually using this url: http://pdfdrm.snu.ac.kr/snudrm/download/PDFReaderSetup.exe