Chapter 2

TRANSACTION LEVEL MODELING
An Abstraction Beyond RTL

Laurent Maillet-Contoz and Frank Ghenassia
STMicroelectronics, France

Abstract: Transaction level modeling (TLM) is put forward as a promising solution above Register Transfer Level (RTL) in the SoC design flow. This chapter formalizes TLM abstractions to offer untimed and timed models to tackle SoC design activities ranging from early software development to architecture analysis and functional verification. The most rewarding benefit of TLM is the veritable hardware/software co-design founded on a unique reference, culminating in reduced time-to-market and comprehensive cross-team design methodology.

Key words: transaction; untimed model; timed model; initiator; target; channel; port; concurrent processes; timing accuracy; data granularity; model of computation; system synchronization; functional delay; annotated model; standalone timed model.

1. THE REVOLUTION

1.1 Call for Raising Abstraction Level

Squeezed by the ever-increasing SoC design complexity, cost, and time-to-market stress, the much-perturbed SoC industry is longing for a solution. The key to this solution is to improve the design productivity through a more reliable design methodology within a shorter design time-frame.

Forwarding critical software development earlier in the SoC design flow is unquestionably helpful to reduce the design cycle time. Such advance implies indeed a hardware/software co-design wherein the software is developed in parallel with the hardware for earlier system integration.

To cope with the rising SoC complexity, a much more rigorous methodology is sought after to assure the reliability of SoC performance at
an earlier stage of the design cycle. A favorable approach is the architecture exploration that analyzes the potential effect of the realistic traffic performed by a system.

Pulling all these factors together, raising the level of abstraction above RTL in the overall SoC design and verification flow has appeared to be a promising solution for the SoC industry.

1.2 Attempts at Raising Abstraction Level

Bear in mind that any attempt made to raise the abstraction level is always a game of balancing the trade-off between the speed and accuracy of a potential simulation model. Our development effort has of course witnessed this game from tip to toe. Before tackling the subject of abstraction level, it is worth considering what the two extreme ends of the SoC design flow could offer.

First, consider the algorithmic model at the highest end of the flow. A complex design usually begins with the development of such a functional model. As an example, a digital signal processing oriented design will have a dataflow simulation engine as its algorithmic model. Since it only captures the algorithm regardless of the implementation details, an algorithmic model has a huge advantage in its high simulation speed. In spite of this, an algorithmic model has no notion of hardware or software component; it models neither registers nor system synchronizations related to SoC architecture. This model therefore cannot fulfill the need of executing the embedded software.

On the other end of the design flow, a pure logic simulation can take place at the register transfer level (RTL). In a conventional SoC logic simulation, RTL models written in hardware description language (HDL) such as VHDL and Verilog are employed as the system hardware. If a processor model is necessary, a design sign-off model (DSM) will typically be used. The advantage of the logic simulation is evidently its great fidelity to the real implementation, i.e. accurate SoC functional and performance analysis. This is nonetheless a price too expensive to pay in terms of the lengthy simulation time. The time consumption has actually further worsened lately due to the high SoC complexity that requires a longer RTL development phase. Moreover, a pure logic simulation cannot execute any software in a reasonable amount of time. A system can only integrate its associated software for observation and analysis rather late in the design flow. Since the breadboard is usually almost ready at this point, any system modification will certainly be too costly at this stage.