Chapter 9

Co-Modeling of Embedded Networks Using SystemC and SDL: From Theory to Practice

ABSTRACT

This chapter gives an overview of a modeling application in the general embedded systems design flow and presents two general approaches for the embedded networks simulation: network modeling and protocol stack modeling. The authors select two widely used modeling languages, which are SDL and SystemC. The analysis shows that both languages have a number of advantages that could be combined by the joint use of SystemC and SDL. Thus, the authors propose an approach for the SystemC and SDL co-modeling. This approach can be used in practice to perform protocol stack simulation as well as simulation of network operation. Therefore, the authors give examples of co-modeling practical applications.

INTRODUCTION

The embedded systems design is a very complicated process. It consists of multiple steps of development which should result in a physical implementation. Some of these steps are closely related to modeling, which simplifies development, reduces its cost and helps to avoid serious bugs and errors and fix them before the implementation in hardware. Modeling can be done

DOI: 10.4018/978-1-4666-6034-2.ch009
by a variety of languages. Among them there are such widely used languages as SDL and SystemC. The main purpose of this chapter is to find out a methodology for embedded systems modeling by SDL and SystemC joint use. To achieve the defined above result we firstly should determine the place of a modeling task in the embedded system design flow and identify main objectives which modeling can solve. Then we find out the possible directions of model’s representation to obtain different simulation results. So finally, we focus on each direction, show their distinctive features and give examples for their application on the basis of the proposed SDL/SystemC co-modeling approach.

EMBEDDED SYSTEMS DESIGN FLOW

An embedded system is a specific combination of computer hardware and software which is specifically designed to perform a particular function (or a range of functions) of a larger system. It usually has strict real-time computational, size and energy restrictions (Heath, 2003), (Barr, 2006), (Kamal, 2008).

There are many implementation steps needed to build an embedded system, and each step is a set of complex actions. Performance modeling helps to understand and establish the major characteristics of the future product. The result of the functional modeling is a specification of the product’s functional behavior. During the design and synthesis step the developers implement the specified mechanisms in details and check them. Validation and verification step ensure that the final implementation behaves in a strong accordance to the specification. All these activities operate on models and not on the real physical object. The reasons for using a model are that, firstly the real product is not available on a development stage, and secondly it is much cheaper to test the specification of a model that on a real device prototype. (Jantsch, 2004).

The embedded systems design encounters a number of difficulties caused by increasing complexity of projects, increasing requirements to products reliability, power consumption and demand to speed-up the project design phase. So the modern approach to the system design implies the parallel execution of some design tasks. It is illustrated by Figure 1 and includes the following stages:

Figure 1. General design flow