Chapter 5

TOP-LEVEL VERIFICATION

After complete formal verification at the block level – based on the techniques presented in the previous chapter – this chapter addresses the verification at the top level. The top-level verification task is required since large systems cannot be handled completely by formal methods due to complexity reasons. Thus, as introduced by the proposed design and verification flow, block-level verification is carried out first and then top-level verification starts on top of the high quality result, i.e. 100% correct proven blocks. Hence, the techniques that are presented in the following focus on the verification of the communication between the proven blocks.

The parts of the proposed design and verification flow that are covered here are depicted in Figure 5.1. In black the top-level parts are shown whereas in light gray the dependencies to the system level and the block level are illustrated, respectively. The dependencies are explained below when the respective top-level task is described.

First, in Section 5.1 an approach for checker generation is introduced. The basic idea is to embed properties directly into the top-level SystemC description after appropriate transformations have been carried out. Then, the embedded properties are checked during simulation at the top level. Especially properties for communication between different blocks are verified using this approach. In Figure 5.1 this verification task is denoted as checker generation. As can be seen for ensuring the verification quality the coverage check at the system level is used. Besides simulation, the checkers are also synthesizable and can be utilized for on-line tests after fabrication. The approach has been published in [GD04, DG05].

In addition to checker generation also formal verification at the top level is investigated for a special class of systems. For embedded systems a formal hardware/software co-verification approach is presented in Section 5.2.
The approach is based on BMC and the formal coverage check to ensure verification quality (see Chapter 4). Besides correctness proofs of the underlying hardware, the hardware/software interface and programs using this interface can be formally verified. The method is shown in Figure 5.1 as the task formal HW/SW co-verification. Parts of the results have published in [GKD05a, GKD05b, GKD06].

5.1 Checker Generation

Besides formal verification of the SystemC blocks as described in the previous chapter, their mutual communication has to be checked. This verification task has to be performed at the top level.

There are several approaches for top-level verification which are based on assertions (for an overview see e.g. [FKL03]). The key idea is to describe expected or unexpected behavior directly in the device under verification. These conditions are checked dynamically during simulation.

To validate properties of SystemC transaction level models a language extension of System Verilog Assertions has been proposed in [EEH+06]. The respective implementation using proxies has been described in [EES+07].