Computers and Electrical Engineering 38 (2012) 1513–1523

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A systematic approach to configurable functional verification of HW IP blocks at transaction level ☆

Tomaž Nahtigal [a], Primož Puhar [b], Andrej Žemva [a,*]

[a] Faculty of Electrical Engineering, University of Ljubljana, Tržaska 25, 1000 Ljubljana, Slovenia
[b] Hella Saturnus Slovenija, Letališka 17, 1001 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 3 March 2011
Received in revised form 14 May 2012
Accepted 14 May 2012
Available online 22 June 2012

ABSTRACT

With demand growing by the day, the complexity of electronic devices is constantly increasing. Since simulation is still the most used approach, functional verification has become one of the major bottlenecks in the design and verification flow.

In this paper we propose a systematic approach to configurable functional verification of electronic devices. Based on a black box approach, it can be applied to any design where behavior can be expressed by a set of functions. It combines simulation- and assertion-based verification into a hybrid verification. The proposed specification-based coverage metric can be configured ranging from a very rapid to an exhaustive verification. The approach uses Transaction Level (TL) modeling to raise the abstraction level, providing faster verification. The results of the proposed design and verification flow, Intellectual Property (IP) and Test Bench (TB) are reusable.

The approach is demonstrated on two case-studies; a video-processing IP block and universal serial bus host controller. The results consider both simulation times and TB generation times.

1. Introduction

Even though electronic devices already combine many different functions, the market continuously demands new functionality. In the future, the functionality of electronic devices will only need to be enhanced, meaning that their complexity will increase drastically, requiring more effort in the design and verification process.

Simulation is still the most used approach to functional verification even though new verification techniques have been developed. However, due to the increased complexity of electronic devices, verification can take up to 80% of the device design time and cost when using Register Transfer Level (RTL) and Hardware (HW) models for simulation ([1–3]). Considering the extreme time-to-market pressures, there is no doubt that advanced new solutions will have to be provided.

One possible solution is in raising the level of abstraction. This approach has been successful in the past (transistor level → gate level → RTL). Raising the level of abstraction to Transaction Level Modeling (TLM) and using SystemC ([4]) to describe the models can provide many benefits. Device behavior can be expressed using high-level functions and high-level data types can be used for communication. The details of device architecture are omitted, allowing for a simpler and faster device model. This also permits a highly systematic approach to device modeling and verification. The designer is allowed to focus on the functionality first and on the implementation second. As a consequence, the verification process is also divided into stages. In the first stage, the high-level functional model is verified. Errors in the functional model that are propagated to

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.
* Corresponding author. Tel.: +386 1 4768 346; fax: +386 1 426 46 30.
E-mail addresses: tomaz.nahtigal@fe.uni-lj.si (T. Nahtigal), primoz.puhar@hella.com (P. Puhar), andrej.zemva@fe.uni-lj.si (A. Žemva).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.05.006

the RTL level, or even to the gate level are very costly to repair in terms of time and money. So it is crucial to find and fix these errors in the early phases of the design and verification process. In the following stage, the RTL model is verified, with more emphasis placed on implementation of the verified functionality.

Additional improvement can be achieved by combining formal Assertion-Based Verification (ABV) with Simulation-Based Verification (SBV), resulting in a hybrid verification. Assertions are a set of formal rules defined by the designer that are checked during the simulation. Assertions can help pinpoint an error in the design and find errors that would otherwise be very difficult to detect.

The models used in this paper are described using SystemC. SystemC allows the HW description in C++, which is the basis for SW/HW co-verification. Therefore, verified functionality can be implemented either in SW or HW.

A key concept in modern digital design is design reuse ([5]). Modeling the device with functional blocks and verifying each block individually allows the designer to construct a database of verified blocks for future use. The Test Bench (TB), constructed for functional verification, can be reused for verification at lower levels of abstraction with the help of translators. The reuse of verified IP blocks and TB results allows for a considerable savings in time and cost.

1.1. Design and verification flow

The design and verification flow shown in Fig. 1 is composed of two parallel flows; design flow and verification flow. The result of the design flow is a TL model and that of the verification flow is a TB.

The basis for an effective design and verification flow is a good specification including, both a design specification and verification specification. The design specification defines the device behavior. The verification specification, also known as the verification plan, defines the verification effort and operating conditions thus helping us to avoid an unnecessary verification effort.

In order to measure the verification success, verification coverage metrics are used. There are as many different coverage metrics as there are verification techniques. We can measure the code coverage, toggle coverage, state coverage, etc. None of these metrics actually checks whether the design meets its specification. The verification of a high-level functional TL model requires a metric that is not based on knowledge of architecture or implementation. We propose a specification-based Functional Coverage Metric (FCM) which measures compliance of the design with its specification. However, even though metrics are a useful indicator, a full coverage does not necessarily mean the absence of errors in the design ([2]). The FCM definition is included in the verification specification. It defines the verification effort and is the basis for TB generation.

The success of a simulation-based verification depends on the quality of the TB used. The easiest to construct and most commonly used is a directed TB, which consists of several directed tests. They are used to test specific corner cases. Though

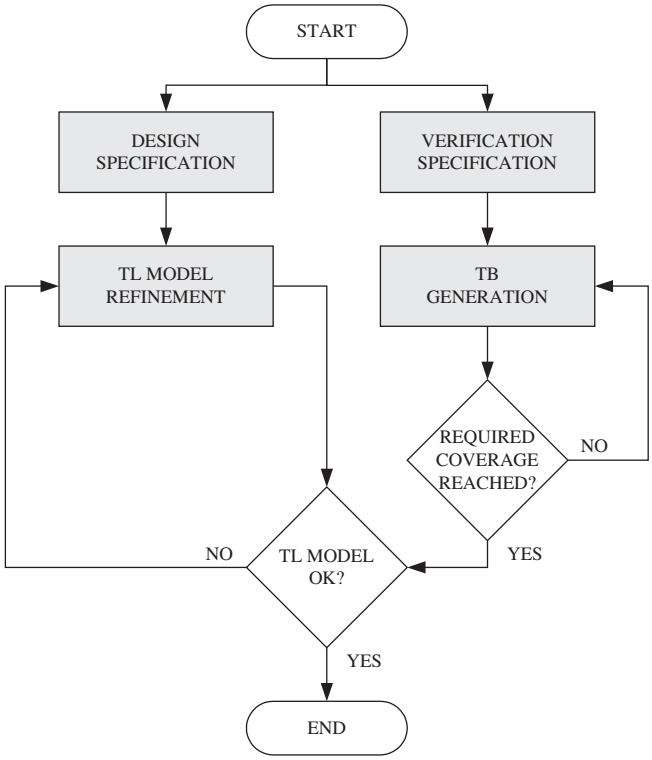

Fig. 1. Design and verification flow.