Available online at www.sciencedirect.com
SCIENCE DIRECT®
INTEGRATION, the VLSI journal 39 (2006) 182–204
INTEGRATION theVLSI journal
www.elsevier.com/locate/vlsi

A PN-based approach to the high-level synthesis of digital systems

Victor R.L. Shen*

Department of Computer Science and Information Engineering, National Formosa University, Huwei, Yunlin 632, Taiwan

Received 20 August 2004; received in revised form 19 May 2005; accepted 30 May 2005

Abstract

A Petri net (PN)-based approach associated with object-oriented technique is proposed to support the specification, analysis, and design of digital systems. Starting from system level to register-transfer level (RTL), the marked Petri net (MPN) with colored tokens is well applied to capture the designer's ideas and to present the system's behavior graphically. Through the net model, reachability analysis technique is employed to formally verify the digital system designed. Hence, using the behavioral properties—*liveness* (i.e. absence of deadlock) and *safety* (i.e. absence of overflow) of the net model can avoid the hardware system from deadlocks and hazards, respectively. From the live and safe MPN model we can obtain the desired hardware prototype at RTL by using the system optimization rules and object-oriented model checking. Furthermore, a time Petri net (TPN) model can be used to check the time consistency among events. This PN-based modeling approach is superior to the current techniques for requirements analysis. Finally, main results are presented in the form of four properties and supported by some experiments.
© 2005 Elsevier B.V. All rights reserved.

Keywords: High-level synthesis; Object-oriented model; Petri nets; Requirements analysis; Rapid protyping

1. Introduction

When developing a top-down design, we usually start from the system level and keep on breaking the design process down until we reach a level where the design can be constructed with

*Tel.: +886 56315570; fax: +886 2 2205 7260.
E-mail addresses: victor.rlshen@msa.hinet.net, rlshen@sunws.nfu.edu.tw.

0167-9260/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2005.05.002

standard off-the-shelf parts or can be synthesized with synthesis tools. Consequently, the requirements specification and specification analysis play a major role among the hardware development stages [1]. In doing so, a correct first step can be taken to keep the designer from wasting his/her energy and cost in the design process. Once the implementation of a digital system is finished, to correct the system will become much harder. Thus, this is the main reason why many researchers are interested in the *executable specification* [2] or *high-level synthesis* [3–9].

The earlier proposed models of synthesis process were mainly related to the interaction of design tools within computer aided design (CAD) frameworks [10], [11]. *Roadmap* [12] model is a data-flow-based model for CAD frameworks with task definition, flow definition, and tool execution. *Ulysses* [13] system provides a blackboard model by using artificial intelligence techniques and a dependency graph called task schema for the design tools execution. *Nelsis* [7] represents a CAD framework associated with design data and design flow management, advanced meta-data handling facility, and consistency management.

In addition, algorithmic state machine (ASM) chart was developed by T.E. Osborne in 1973 [14]. Since then it has been widely applied to assist the designer in expressing the abstract algorithm and to support the conversion of the algorithm into hardware. However, it does not provide a straightforward way to eliminate deadlocks and hazards in the digital system. This motivates us to propose an attractive alternative to solve the problems.

In this paper, the digital system is specified with a marked Petri net (MPN) model (also called *marked graph*) and a time Petri net (TPN) model. Through the MPN model, two behavioral properties including *liveness* and *safety* can be exploited. According to Ref. [15], an example of *liveness* property can be stated as follows: ''for every path in the future, if there has been a *Request* signal, then eventually there will be an *Acknowledge* signal in response to the request on at least one node on the path.'' And an example of *safety* property can be stated as follows: ''for every path in the future, at every node on the path, if the *Request* is low, it must remain low until *Acknowledge* goes low.''

Liveness and safety properties are even crucial to the incomplete specifications [16]. If we can demonstrate that the two properties are true, then we call a digital system design correct with respect to the above two properties. By using those behavioral properties, hardware problems at register-transfer level (RTL) like *deadlocks* and *hazards* can be avoided. Wherein, deadlocks mean that a transition is not enabled forever; and hazards might occur at transitions where their input places have differently colored tokens. Moreover, by using the TPN model, the timing consistency among events (i.e. transitions) can be efficiently checked.

To begin with, the designer can just concentrate on the digital system behavior and specify it with an MPN model. In other words, the behavior of MPN must be as close to the digital system requirements as possible. Then the MPN model is checked whether its liveness and safety are met. The internal structure of a digital system will be decomposed iteratively until satisfaction. This approach is a formal method whose specification technique is easy to understand, and whose analysis is automated.

In this design automation system, an object-oriented technique [17] would also aim to provide us with a user-friendly interface for heterogeneous design data, reuse of existing design information, hierarchical design methods, management of different versions of a design, and an integrated environment of CAD tools.

Download English Version:

https://daneshyari.com/en/article/543001

Download Persian Version:

https://daneshyari.com/article/543001

Daneshyari.com