Microprocessors and Microsystems 39 (2015) 1245–1254

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A standardized design methodology for complex digital logic components of cyber-physical systems

F. Chen [a], H. Ye [a], J. Yang [a], Y. Huang [b,*], J. Zhang [c], X. Qi [a], C. Zhao [a], J. Zhu [a], W. Zhou [d]

[a] Anhui Normal University, China
[b] Anhui Science and Technology University, China
[c] University of Southern Queensland, Australia
[d] Wuhan University, China

ARTICLE INFO

Article history:
Received 25 April 2015
Revised 12 August 2015
Accepted 15 August 2015
Available online 8 September 2015

Keywords:
CPS
Digital logic
Modeling
Effectiveness
Verification
Component

ABSTRACT

As an important part of cyber-physical systems, the digital logic system's complexity are rapidly increasing, and its design flows become more and more tedious. A modeling and verification methodology for complex digital logic components is presented to improve the development quality and efficiency. In order to help developer to understand the design intent preferably and speed up the development process, design activities are carried out under a convenient modeling methodology and a precise verification solution for completing the design cycle. Its calculus system offers a theoretic way to connect components via connectors, and then provides a theoretical basis for further verification. A washing machine controller design shows that the modeling can reflect the design intent of the designers effectively, detect some design errors which maybe result in modeling failures in implementation as soon as possible, and avoid the negligence and errors in modeling for complex digital systems.

1. Introduction

With the development of computer theory and technology, more and more smart digital logic systems are applied in many cyber-physical system (CPS) related fields, e.g. automobiles electronics, hand-held game systems, industrial environments, telecommunication or fabrication equipments, aircraft electronics, medical systems, military applications, authentication systems, consumer electronics, smart buildings, robotics, etc [1–5]. As a multi-dimension complex system integrated with computing, networking, and physical environment, realizes real time perception and dynamic control of the physical devices or the environment through computing, communication and control. Therefore, it emphasizes the process of feedback and control [6].

Compared with the software implementation methods [7,8], the digital logic implementation helps to improve the performance and save the size. In order to achieve that process, a CPS is usually composed of different components including some digital logic components such as adders, registers, processors, memory, interface circuits, sensors, controllers, and so on. In the development of complex digital logic systems, before we start out designing it and submit our design result for manufacturing, usually we have to face a key question, that is, how to ensure that our design intent is consistent with the requirements.

Component-based modeling is a method for constructing complex digital logic systems between the requirements and actual systems production. Through defining some components, designers can model a virtual application system. One of our contribution is to introduce an efficient simple method to model complex digital logic system quickly and make the specification and verification process practical for real designs. Complex digital logic circuit can be divided into combinational digital systems and sequential digital systems, corresponding to combinational logic components and sequential logic components which belong to atomic components, and namely cannot be divided to smaller components. A bigger component which is a compound logic component is constructed from those smaller components connecting with connectors (wires). According to the different complex digital logic components, this paper proposes the rules and their implementation methods for the effectiveness verification. There are two key criterions- the first one is that the specification of a component is complete; the second one is that a component should be stable. They can be used to verify the effectiveness for complex digital logic components so as to ensure that the model can reflect the design intent of the designers effectively, detect some design errors which maybe result in the model failures in implementation as

* Corresponding author at: Department of Computer Science, Anhui Science and Technology University, Bengbu, Anhui 233100, China. Tel.: +86 18226703193.
E-mail address: hhyyjsh@163.com (Y. Huang).

http://dx.doi.org/10.1016/j.micpro.2015.08.012
0141-9331/

soon as possible, and avoid the negligence and errors in modeling for complex digital logic systems with components.

In this paper, some related works are introduced in Section 2. Sections 3–5 respectively address modeling and verification for combinational, sequential and compound logic components. In Section 6, a washing machine controller is designed in the component-based modeling and verification methodology as a case.

2. Related works

Modeling plays a central role in systems engineering. Modeling is a process to discover the system requirements, and its purpose is to associate the structure design with the system behaviors, then control the system architecture visually [9–11]. It can profitably replace experimentation on actual systems, and provide a basis for rigorous system development and implementation. During modeling, the most important one task is to specify what we want, and the second one is to make sure that what we specify is what we want [12]. Specification and verification have become the most important two tasks in current design flows, and they have major impact on the timely delivery of a functionally correct product. In the development of complex digital logic systems, before we submit our design results for manufacturing, usually we have to face a key question-how to ensure whether our design result is consistent with the requirements. Experience has showed that more than 40% of the human resources and 70% of the computing resources in micro-processor design project are devoted to the verification task. Verification has become the most important task in the current design processes, and it has a significant impact on the timely delivery of a functionally correct product [13].

Lots of efforts are put into research on theories and methods of modeling for complex digital logic systems. However how to reduce the design cost, time and system complexity effectively is a permanent problem which has attracted many scholars. Component-based modeling method plays a vital role in the design of digital logic systems, can model interaction behaviors, and brings into high efficiency and re-usability in an abstract layered model [14]. Pfeifer [15] provided SimConnect and SimTalk which enable heterogeneous, distributed, hardware/software co-simulation and emphasize the simulation of software interacting with simulated world-model electrical, mechanical, and physical effects. Davare [16] presented METROII which supports platform-based design for heterogeneous embedded components. The Ptolemy project [17] directed by Edward A. Lee in UC Berkeley, studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components. But this method is only operable in TTA (Time Trigger Algorithm), not in ETA (Event Trigger Algorithm). So a major problem area being addressed is the use of heterogeneous mixtures of models of computation. MIC (Model Integrated Computing) [18] of Vanderbilt University addresses the problems of developing software integrated systems by providing rich, domain-specific modeling environments including model analysis and model-based program synthesis tools, which is used to create and evolve integrated, multiple-aspect models using concepts, relations, and model composition principles routinely used in the specific field, to facilitate systems/software engineering analysis of the models, and to automatically synthesize applications from the models. MIC has been used to develop many different technologies and solutions for industry and government.

The verification of digital logic systems has been also thoroughly studied. However how to save the design cost effectively is a permanent problem. The current verification methods include software simulation [19], formal verification [13] and FPGA verification. Many software simulation tools such as ModelSim, Isim, Simulink and HyperLynx have been developed and widely applied. Unfortunately the result of these methods is not necessarily reliable. Formal verification can be divided into three categories: equivalence checking, model checking and theorem proving. Many EDA tools such as Formality, FormalProTM and Affirma can be used to do equivalence checking [20]. However this method has certainly limitations that the time of element storage is too long and the ability to deal with large-scale circuit is not strong [21,22]. Zhao put forward a SAT unbounded model-checking framework based on CNF [23]. Theorem proving is lightly used for its high cost. Program can be downloaded into the FPGA devices to achieve ICV(In-Circuit Verification). Unfortunately the FPGA devices have limited external pins and it can hardly monitor internal signals. Although it can solve the problem of signal transmission through serial transmission and parallel transmission conversion, the transmission speed limits the efficiency of verification.

3. Combinational logic components

3.1. Basic definitions

Definition 3.1 (*Port*). A port is denoted as p or $p_{MSB...LSB}$, and a set of ps is denoted as P. A port is a connection interface among components. It can be used to receive input digital signals and send output digital signals.

Definition 3.2 (*Input port*). An input port is denoted as ip or $ip_{MSB...LSB}$, and a set of ips is denoted as IP. An input port is used to receive input digital signals.

Definition 3.3 (*Output port*). An output port is denoted as op or $op_{MSB...LSB}$, and a set of ops is denoted as OP. An output port is used to receive output digital signals.

Definition 3.4 (*Truth value*). A truth value is denoted as v and a set of vs is denoted as V. Truth values include **TRUE**(**1**), **FALSE**(**0**), **HIGN-IMPEDANCE**(**z**, **Z**), and **UNKNOWN**(**x**, **X**). The truth values for IP is called input vector, and it is abbreviated as IV and a set of all the possible IVs is denoted as Σ. The truth values for OP is called output vector, and it is abbreviated as OV and a set of all the possible OVs is denoted as Γ.

Definition 3.5 (*Value*). A value is denoted as v and a set of vs is denoted as V. Values include truth values, integer values and real values. The value for IP is called input vector, and it is abbreviated as IV and a set of all the possible IVs is denoted as Σ. The values for OP is called output vector, and it is abbreviated as OV and a set of all the possible OVs is denoted as Γ.

Definition 3.6 (*Implicant term*). An implicant term is an item $\langle IV\rangle : \langle OV\rangle$.

Definition 3.7 (*Minterm*). A minterm is an item $\langle IV\rangle : \langle OV\rangle$ in which the truth value in each IV should only be **TRUE**, **FALSE** or **HIGN-IMPEDANCE**(**z**, **Z**) can only appear one time.

Definition 3.8 (*Implicant minterm*). An implicant minterm is denoted as $[\langle IV\rangle : \langle OV\rangle]$ and the IV should be a minterm.