Specification and Validation of Control Intensive ICs in hopCP

VENKATESH AKELLA
GANESH GOPALAKRISHNAN

(akella@cs.utah.edu)
(ganesh@bliss.utah.edu)

Dept. of Computer Science
University of Utah
Salt Lake City, Utah 84112

Keywords: asynchrony, behavioral simulation, formal methods, hardware description languages, formal specification and validation

Abstract. Control intensive ICs pose a significant challenge to the users of formal methods in designing hardware. These ICs have to support a wide variety of requirements including synchronous and asynchronous operations, polling and interrupt-driven modes of operation, multiple concurrent threads of execution, non-trivial computational requirements, and program modifiability. In this paper, we illustrate the use of formal methods in the design of a control intensive IC called the “Intel 8251” Universal Synchronous/Asynchronous Receiver/Transmitter (USART), using our hardware description language ‘hopCP’. A feature of hopCP is that it supports communication via asynchronous ports in addition to synchronous message passing. Asynchronous ports are distributed shared variables writable by exactly one process. We show the usefulness of this combination of communication constructs. We outline algorithms to determine safe usages of asynchronous ports, and also to discover other static properties of the specification. We discuss a compiled-code concurrent functional simulator called CFSIM, as well as the use of concurrent testers for driving CFSIM. The use of a semantically well specified and simple language, and the associated analysis/simulation tools helps conquer the complexity of specifying and validating control intensive ICs.

1 Introduction

Over the last two decades, VLSI technology has advanced by leaps and bounds, and has contributed to a rapidly increasing performance/price ratio of hardware. With these improvements, however, have come a variety of new problems. Although the speed and the scale of VLSI systems continues to grow, their functional complexity may not scale at the same rate, unless some of the problems that have begun to creep up at the level of system design are properly tackled and solved.

There are many sources for the problems encountered at the system level of hardware design. Many of these are problems of scale akin to those found in the design of large software systems. The more serious of these problems are, however, due to the concurrent nature of hardware, and because of the large number of complex features that hardware designers are trying to support in VLSI systems they are currently building.

We can illustrate many of the above mentioned problems, as well as possible solutions, through one example: the Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) [23]. Integrated circuits (ICs) such as the 8251 USART exhibit diverse behaviors. They typically possess independent threads of execution, have coexistent synchronous (clocked) and asynchronous (unclocked) subcomponents, support multiple modes of operation, such as the interrupt-driven and the polled modes. They are programmable to set the baud rate, the number of stop bits, start bits, error flags, and the synchronization scheme etc.. They can perform computations, such as error-checking, assembling and disassembling of data, and code-conversion. Such ICs are commonly classified as “control intensive”. There are very many control intensive ICs in day-to-day use today; we selected the 8251 because it has been widely used in the past as a benchmark for comparing