Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Characterizing asynchronous variable latencies through probability distribution functions

J.M. Colmenar [a], O. Garnica [b,*], J. Lanchares [b], J.I. Hidalgo [b]

[a] *C.E.S. Felipe II. Complutense University of Madrid, Spain*
[b] *Architecture and Technology of Computing Systems Group (ArTeCS), Complutense University of Madrid, Spain*

ARTICLE INFO

Article history:
Available online 7 October 2009

Keywords:
Asynchronous
Microprocessors
Variable latency
Probability distribution functions

ABSTRACT

Asynchronous systems are attracting the interest of the designer community because of several useful features for sub-micron technologies: process-variation tolerant, low-power, removal of the clock tree generation, etc. One of the main problems for the simulation of these systems is the variable computation delays of their modules, that compute as fast as possible under the actual conditions of the system. This behavior complicates the high-level simulation of such systems and it is the main reason for the lack of simulation tools devoted to asynchronous microarchitectures. In this paper we present a modeling method useful for this kind of systems that describes the variable computation delay of an asynchronous circuit by using probability distribution functions. This method is deployed in an architectural simulator of a 64-bit superscalar asynchronous microarchitecture where the computation delay of each one of the modules of the microarchitecture was characterized through a probability distribution function. The experimental results show that the asynchronous behavior is successfully modeled, and the architectural simulations of standard benchmarks is affordable in terms of wall-clock simulation time.

1. Introduction

Each successive technology is aggravating the well-known inconvenience of synchronous systems, that is, the need of over-designing the system to satisfy design constraints – about performance, power-consumption or device reliability under any corner conditions of process, voltage, temperature and on-chip-variations – in order to meet yield rates. Traditional techniques for minimizing design exposure to process and environmental variations are quickly becoming difficult to implement and consume an increasingly large portion of the microprocessor design.

As a result, the interest on asynchronous systems in the community of circuit designers is growing, *i.e.* [1,2]. These circuits have a number of interesting inherent properties that solve some of the problems of synchronous designs:

- High performance [3,4]: the global circuit performance of fully asynchronous systems corresponds to the performance of the average case. In asynchronous systems a new computation starts immediately after the previous has finished [5].
- Robustness towards variations on supply voltage, temperature and fabrication process [6,7]: the functionality is designed to be independent from the timing, which allows the circuit to compute as fast as possible under any temperature, process and voltage corner.
- Modular design [8,9]: the local timing and the communication protocol interfaces allow designers to create modular systems, even based on templates or asynchronous IP cores.
- Absence of clock distribution problems: there is no global clock signal in the system.

Nevertheless, there are two main drawbacks when designing asynchronous systems.

First, the control logic that implements the handshaking between asynchronous circuits usually represents an overhead in terms of silicon area, delay and power consumption. But, as shown in [3,10], the overhead of the control logic may be hidden or compensated.

Second, there is a lack of CAD tools devoted to asynchronous circuits. Despite many CAD tools and algorithms for synthesis of asynchronous systems, *i.e.* [11–13], are currently available, there are few tools related to architectural simulation of asynchronous systems.

One of the main obstacles for the architectural simulation of asynchronous circuits and, therefore, for the development of simulation tools, is the data-dependency of the computation

* Corresponding author.
E-mail addresses: jmcolmenar@cesfelipesegundo.com (J.M. Colmenar), ogarnica@dacya.ucm.es (O. Garnica), julandan@dacya.ucm.es (J. Lanchares), hidalgo@dacya.ucm.es (J.I. Hidalgo).

0141-9331/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2009.09.005

delays. The delay due to the computation may be different for each incoming data on an asynchronous circuit because it computes as fast as possible without any timing constraint. Up to our knowledge, there are not reported methods in literature related to the data-dependent characterization of the variable computation delay applied to the architectural simulations of asynchronous processors.

The simulation of this kind of systems requires firstly a method that enables, in a cost-effective way, both in terms of memory requirements and computing power of the simulation infrastructure, the characterization of asynchronous modules with data-dependent computation delays; and secondly, a tool able to simulate a processor whose modules are asynchronous circuits.

Hence, the main contributions of this paper are:

1. A modeling method based on probability distribution functions (PDF) that allows the cost-effective architectural simulation of complex asynchronous systems, and
2. A tool that allows the simulation of an asynchronous Alpha 21264-like processor. This tool deploys the modeling method based on PDFs and permits to configure most of the parameters of the processor with the aim of studying the processor performance under standard workloads, typically SPEC2000 suite.

The rest of the paper is organized as follows. In Section 2 we review works focused on measuring the performance of asynchronous systems. In Section 3 we introduce the problem of characterizing and simulating the data-dependent behavior of asynchronous circuits. In Section 4 we detail the method followed to obtain the PDF from a sample of delays, and we show a statistical metric to prove the quality of the sample. In Section 5 we describe the implementation of the PDF characterization within an architectural simulator, and in Section 6 we verify the cost-effective and successful architectural simulation of a superscalar asynchronous microarchitecture. Finally, in Section 7 we present the conclusions and the future work.

2. Related work

PDFs, powerful statistical tools able to describe the probability of a given variable taking different values, may summarize the variable computation delay of an asynchronous module.

The use of PDFs is widespread in science: many natural phenomena can be modeled by using distribution functions. For instance, in relation to the energy state of a particle, three different distribution functions have been described [14]: Maxwell-Boltzmann, Bose–Einstein and Fermi–Dirac distributions; and many other research areas also trust in distribution functions to model different parameters: level fluctuations in water channels [15]; clouds distribution [16]; building simulations [17]; or satellite validations [18].

In the field of computer science, PDFs have been used mainly to model the behavior of packet traffic within TCP/IP networks [19–21], and in other areas of computer science like the systems design, the use of PDFs is not very common up to our best knowledge, although the performance analysis of asynchronous systems has taken advantage of other probabilistic techniques. Next we present, in chronological order of publishing, works related to this topic.

In [22–24] Xie and Beerel present different techniques to analyze the performance of asynchronous systems. The authors use high-level representations of the circuits, from Probabilistic Finite State Machines to Stochastic Petri Nets, STPN.[1] These descriptions use symbolic expression to describe PDFs – typically, the exponential random distribution – so it is required to obtain them for every node in the circuit. This requirement is the main drawback of this technique because of the difficulty to obtain such a representation in complex circuits.

In [25] Chakraborty et al. present a different approach to analyze the delay for asynchronous systems. They describe the circuit by Marked Graphs,[2] and apply Monte Carlo simulations to obtain the mean and variance that characterize the behavior of each system element. The result is a range of values for the system timing mean and variance. Therefore, the aim of this analysis, is to verify the system timing, and not to estimate system performance. The main drawback is the need for a large number of simulations due to the Monte Carlo approach.

In [27], McGee et al. describe a method to estimate the asymptotic performance based on Marked Graphs in which the variations in the input data arrival times and the computation delays of the system components are modeled using PDFs. However, this method has two restrictions: firstly it only allows the use of exponential random distribution, unsuitable to describe the behavior of many asynchronous circuits; secondly, the method is limited to decision-free systems, limiting its extension to complex microarchitectures, as the authors state in their work.

Finally, up to our knowledge none of the microarchitectural simulators of asynchronous processors developed up to date [28–30] use PDFs to estimate the performance of such a system.

In summary, PDFs have been primarily used to build high-level descriptions of asynchronous circuits, and not to estimate their performance. In addition, they have commonly been described using a well-known random distribution, *i.e.* exponential random distribution, that enables the use of powerful analysis techniques but limiting the type of real-life circuits that can be analyzed, *i.e.* preventing PDF particularization to discrete functions.

3. Simulating data-dependent delays

The timing of an asynchronous circuit is not homogeneous because its computation delay depends on the data being processed. Furthermore, considering a complex asynchronous system formed by modules that compute in an independent way – every individual module of the system presents its own data-dependent computation delay – and perform the communication of results between them using a handshake protocol, the characterization of the computation delay of the whole system becomes more difficult.

One trivial solution for the problem of simulating the data-dependent computation delays in a complex asynchronous system may have the following steps: (1) obtain the computation delays for all the different input data of each module of the asynchronous system (through gate-level timing simulations of each module, for instance); (2) store the obtained computation delays in convenient structures; (3) use the computation delays in simulations of the whole asynchronous system, matching each data with its computation delay on each module and modeling the handshake protocol. This solution is not affordable for several reasons:

- High number of simulations: all possible input data for all the modules of the asynchronous system should be simulated due to the data-delay matching.
- Long simulation times: wall-clock time for gate-level simulations is usually very long.

[1] STPNs are Petri Nets that add nondeterministic time through adjustable randomness of the transitions.

[2] A Marked Graph is a subclass of Petri Nets in which every place has one incoming arc, and one outgoing arc. This means, that there cannot be conflict (decision-free systems, [26]), but there can be concurrency.