Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-25 (2016) 092–097

A SystemC Based Framework for Cycle Accurate Processor Simulation and Parameter Analysis

Johannes Kohl * **Marc Reichenbach** * **Carsten Demel** * **Dietmar Fey** *

* *Department of Computer Science, Chair of Computer Architecture Friedrich-Alexander-University Erlangen-Nuremberg, Germany (email{Johannes.Kohl, Marc.Reichenbach, Carsten.Demel, Dietmar.Fey}@fau.de)*

Abstract: The development of new domain specific microprocessors architectures is a very complex task. Hardware designers are confronted with several hundred design parameters (e.g. cache size, ALU operation, etc.) which can be combined in a nearly arbitrary way to develop new architectures. Often these parameters are guessed by the designer or *best practice* is used to find an apparently good parameter set. The utilization of virtual hardware can be one solution to solve this problem. Different parameter sets can be evaluated inside a simulation environment, without the need for designing any real hardware. Unfortunately, most virtual hardware simulators are either to inflexible or not accurate enough to get valuable results. Therefore, this paper presents a SystemC based framework, which enables the construction of highly parameterizable virtual hardware processor models, directed at parameter analysis. The structure and working principle of two exemplary processors are showcasing the use of this framework. A presentation of how different parameter sets impact the programs execution time and power consumption of the processor will be given, as a result of the virtual hardwares execution.

Keywords: Design Tools and Application Software, Emulators and Simulators, Microprocessor Based Control Systems, Cycle Accurate Processor Simulator, Design Space Exploration

1. INTRODUCTION

The development of fast and energy efficient microprocessors is a very important task in todays automated world. New architectures that are robust, fault-tolerant and capable to handle hard real-time applications have to be created. But how to find an optimal processor architecture for domain specific applications is still an open question. Often processors are designed by applying *best practice*, but for the final evaluation a prototype has to be created. If a parameter has to be changed after evaluation, the very time and cost consuming design process has to be repeated. Therefore, an early and exact determination of eminently suitable parameters for the processor architecture is needed, already before hardware development starts.

In principle, there are two different ways to determine architectural parameters in a very early design stage. The first is the analytical, or mathematical way where an architecture is parameterized and described with mathematical equations. By solving this equations, optimal parameter sets can be derived. However, this approach is not very convenient, as architectures, like multiprocessor systems, become more complex and the equation system exceeds the possibility to be solved in reasonable time. A second approach is, to create a virtual hardware (or processor) model and running it with real input data while counting clock cycles, cache misses etc. in the simulation. This approach is very common in current research and in industry, but lacks regarding flexibility: Either the simulation environments are to abstract esp. Instruction-Set Simulators (ISS) like OVP [1] (Open Virtual Platform) and therefore give insufficient results. Or, they are tailored to a very detailed level of exactly one architecture, which can not be changed and does not apply to the optimization problem's domain. Therefore, a new simulation framework has to be developed to combine the advantages of both simulation approaches.

In this paper, we introduce PACAS (Parameter Analysis Cycle Accurate Simulator), a SystemC based framework for flexible processor design and analysis, as well as evaluation of different design parameters. The main features of this framework are the generation of highly configurable processor models with custom architectures to test and optimize billions of different parameter variations. PACAS features adaptability for different processor architectures through abstracting framework elements like pipeline stages, caches and registers. A high traceability is achieved through a hierarchical class system which structures the architecture and allows to count bit toggles of every register in a processor to allow an early power estimation. High simulation speed is accomplished through an

⋆ The authors greatfully acknowledge the support by profichip GmbH and the Bayrische Staatsministerium für Wirtschaft Infrastruktur und Technologie (StMWIVT), in the context of the R&D program IuK Bayern under Grant No. IUK-1308-0009.

[1] http://www.ovpworld.org/

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.12.016

abstract description of the architecture and the framework components if no traceability is needed.

This paper is structured as follows. In this Section the idea of creating a new simulation framework has been discussed. In the next section different approaches of other simulation techniques are evaluated. The basic structure of the simulation framework and different parts of PACAS are explained in section 3 while section 4 showcases how different processors can be created utilizing PACAS. An evaluation of the processors with different parameters set, demonstrating the flexibility and capability of the framework, is given in section 5. Finally a conclusion and an overview about future work is given.

2. RELATED WORK

Several kinds of processor architecture modeling techniques are used in academic research, ranging from the most abstract ADL (Architecture Description Language), which might not even support architecture simulation like xADL 2.0 presented in Dashofy et al. (2001), to the more accurate CAS (Cycle Accurate Simulator), that can be precise enough to estimate even the architectures power consumption. However, all these different types have their particular strengths and weaknesses. The following will discuss different simulation methodologies and their subsequent implications for parameter analysis.

2.1 Instruction Set Simulators

Starting with ISS, different implementations of varying detail can be found. The implemented model of these architecture simulations ranges from event based modeling Luckham and Vera (1995-09) and timed automata Wang et al. (2011) to advanced simulation models like Sniper Carlson et al. (2011), a x86 multi core simulator which tackle the Multi-Core scalability problem.

While there are SystemC based ADL simulators, like the one presented in Fummi et al. (2004), these ISS all have the downside of only being accurate to the instruction level. This implies, that the actual effort, to perform the instructions, can not be evaluated, e.g. in terms of power usage. Therefore, all ISS alike lack the possibility for deep parameter analysis and consequently do not suffice the requirements for multidimensional DSE (Design Space Exploration).

2.2 Mathematical processor models

Other popular approaches are mathematical and statistical models. These models range from simple mathematical equations for specific parts of the design like caches Berg et al. (2006) to full featured, complex processor models with dozens of formulas. This comes especially evident when parallel or shared-processor models have to be described like shown in Yashkov (1992) and Hong and Kim (2009). The problem with all these mathematical models is, that the more accurate the model has to be, the harder and more complex it gets to formulate the equation, which also results in a very time consuming development phase. Moreover, every design change inherits the risk of complete invalidation of the postulated formula and therefore asks for starting all over again.

Most mathematical models like formal processor models shown in Wilding et al. (1998); Hardin et al. (1999) are

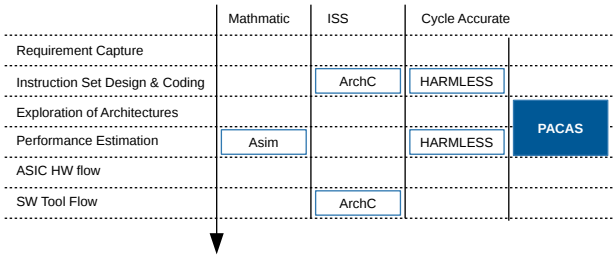

Fig. 1. Application field, within the design flow described in Nurmi (2007), of exemplary simulation techniques and tools compared to the PACAS framework.

directed at performance evaluation like the by Emer et al. presented Asim Emer et al. (2002). However, Reichenbach et al. showed in Reichenbach et al. (2011) the process of formulating such a model and the enormous effort behind it becomes evident. The Non-Functional Properties of the design can not be regarded and therefore a deeper DSE with power consumption and clock rate values is not possible.

2.3 Cycle Accurate Simulators

Despite real CAS more and more ADLs with ISS base protrude into the CAS research field, like cycle approximation Franke (2008), which requires prior training and therefore is fixed to a specific architecture or frameworks which claim to have mastery of both, ISS and CAS, like HARMLESS Kassem et al. (2009, 2008) and ArchC Rigo et al. (2004). These simulators have their strengths in fast and flawless simulator generation, However their underlying simulation model is not set to a specific standard, e.g. not forced to be cycle accurate, and, despite their parameter fixed design methodology, the real achievable toggle accuracy is very limited. In this very same category falls Yeh et al. (2010) which is fixed to preset architectures and directed at SoC (System on Chip) simulations.

While fully featured CAS like LISA Hoffmann et al. (2001), with its just-in-time compiler Braun et al. (2004), and equally gem5 Binkert et al. (2011), with its extension McPAT Li et al. (2009), have a very extensive feature set, even supporting peripheral simulation, they are not suited for every applicable case. LISA e.g. does not support precompiled, parameterizable simulation models, which limits the simulation speed by requiring the additional step of compilation and therefore is not applicable for large scale DSE. Gem5, on the other hand, only supports a fixed set of predefined architectures, with very strict and preset pipeline designs. Those strict architectural presets do not always fit the needs of domain specific processors, especially in respect to unconventional and uncommon processor architectures like PLCs.

As has been shown, most ADLs with simulator backend lack the accuracy needed for effective DSE, especially when *Non-Functional Properties* like area, clock rate and power consumption are of interest, or are too inflexible when widespread parameter analysis, with multiple components and very divergent parameter sets, is asked for. Therefore their utilization for a framework directed at parameter analysis is not suitable.

Fig. 1 compares the main field of application within the processor architecture development based on the design flow described in (Nurmi, 2007, p.70) of exemplary estab-